ROCKWELL SOFTWARE

Arena®

User's Guide

Doc ID ARENA-UM001A-EN-P

Rockwell Automation

Contacting Rockwell Software

Technical Support Telephone—1.440.646.5800
Technical Support Fax—1.440.646.5801
World Wide Web—www.software.rockwell.com

Copyright Notice

© 2004 Rockwell Software Inc., a Rockwell Automation company. All rights reserved. Printed in the United States of America.

This manual and any accompanying Rockwell Software products are copyrighted by Rockwell Automation. Any reproduction and/or distribution without prior written consent from Rockwell Automation. is strictly prohibited. Please refer to the license agreement for details.

Commercial runtime models may be legally loaded and run only by employees of organizations with a commercial Arena license. Models created using a research licenses may not be used for commercial use. Any other use of a runtime model is illegal and unauthorized.

Commercial Arena software can be obtained by contacting Rockwell Software at 1.412.741.3727 or contacting your local representative (listed under partners at www.ArenaSimulation.com <http://www.ArenaSimulation.com>).

Trademark Notice

Arena and SIMAN are registered trademarks and the phrase “Forward Visibility for Your Business” and the Rockwell Software logo are trademarks of Rockwell Software Inc., a Rockwell Automation company.

Microsoft, Active X, Visio, and Visual Basic are registered trademarks of the Microsoft Corporation.

All other trademarks and registered trademarks are the property of their respective holders and are hereby acknowledged.

Warranty

This Rockwell Software product is warranted in accord with the product license. The product’s performance will be affected by system configuration, the application being performed, operator control, and other related factors.

This product’s implementation may vary among users.

This manual is as up-to-date as possible at the time of printing; however, the accompanying software may have changed since that time. Rockwell Software reserves the right to change any information contained in this manual or the software at anytime without prior notice.

The instructions in this manual do not claim to cover all the details or variations in the equipment, procedure, or process described, nor to provide directions for meeting every possible contingency during installation, operation, or maintenance.

Contents

1 • Welcome 1

What is Arena software? . . . 1
Intended audience. . . . 1
Where can I go for help? . . . 2
Reference the user's guides . . . 2
Explore our examples . . . 2
Get help. . . . 2
Use the SMARTs library . . . 2
Access the Arena Symbol Factory . . . 3
Get phone support. . . . 3
Get Web support . . . 3
Get training . . . 4
Get consulting services . . . 4
Contact us . . . 4

2 • Getting Started 5

Introduction . . . 5
Our task: Analyze a home mortgage application process . . . 5
The Arena modeling environment . . . 6
Map your process in a flowchart . . . 7
Create the mortgage application entities . . . 7
Process the applications . . . 8
Decide whether applications are complete . . . 10
Dispose the applications to terminate the process . . . 10
What is a module? . . . 11
Define model data . . . 12
Initiate mortgage application (Create module) . . . 12
Review application (Process module) . . . 13
What are entities? . . . 13
Complete? (Decide module) . . . 15
Accepted, Returned (Dispose module) . . . 16
Mortgage review clerk (Resource module) . . . 17
Prepare for the simulation. . . . 18
Save the simulation model . . . 18
Simulate the process. . . . 18
View simulation reports . . . 20
Enhance the visualization process . . . 21

Animate the mortgage review clerk resource . . . 22
Plot the number of applications in-process. . . . 23
Rerun the simulation. . . . 24
Next steps . . . 24

3 • The Basic Process Panel 27

Flowchart modules . . . 27
Create . . . 27
Dispose . . . 28
Process . . . 29
Process — Resource dialog . . . 31
Decide . . . 32
Batch . . . 33
Separate . . . 34
Assign . . . 35
Record . . . 36
Data modules . . . 38
Entity . . . 38
Queue . . . 39
Resource . . . 40
Variable . . . 41
Schedule . . . 42
Set . . . 44
Calendar schedule information . . . 45
Time Pattern . . . 45
Calendar Exceptions . . . 47
Calendar States . . . 48

4 • The Advanced Process Panel 49

Flowchart modules . . . 49
Delay . . . 49
Dropoff . . . 50
Hold . . . 50
Match . . . 52
Pickup . . . 53
ReadWrite . . . 53
Release . . . 55
Remove . . . 56
Seize . . . 57
Search . . . 59
Signal . . . 60

Store . . . 61
Unstore . . . 61
Adjust Variable . . . 62
Data modules . . . 63
Advanced Set . . . 63
Expression . . . 64
Failure . . . 65
File . . . 66
StateSet . . . 67
Statistic . . . 68
Storage . . . 70

5 • The Advanced Transfer Panel 71

General flowchart modules . . . 71
Enter . . . 71
Leave . . . 73
PickStation . . . 75
PickStation — Stations dialog . . . 76
Route . . . 77
Station . . . 78
Conveyor flowchart modules . . . 79
Access . . . 79
Convey . . . 80
Exit . . . 81
Start . . . 82
Stop . . . 82
Transporter flowchart modules . . . 83
Activate . . . 83
Allocate . . . 83
Free . . . 85
Halt . . . 85
Move . . . 86
Request . . . 87
Transport . . . 88
Data modules . . . 90
Sequence . . . 90
Conveyor . . . 92
Segment . . . 93
Transporter . . . 93
Distance . . . 95
Network . . . 96

Network Link 96
Activity Area 97

6 • The Flow Process Panel 99

Flowchart modules 99
Tank 99
Sensor 101
Flow 102
Regulate 104
Seize Regulator 105
Release Regulator 106
Data module 107
Regulator Set 107

A • Statistical Distributions 109

Beta 111
Continuous 112
Discrete 114
Erlang 115
Exponential 116
Gamma 117
Johnson 118
Lognormal 119
Normal 120
Poisson 121
Triangular 122
Uniform 123
Weibull 124

B • Library Shapes and Symbols 127

Arena .plb picture files 127
Arena .plb libraries 128
Arena Symbol Factory 128
The Symbol Factory application 128

C • Using Visio and Arena 131

Introduction 131
The Visio Process Simulation add-in 131
Start a drawing 131

Place and connect shapes 133
Enter custom properties 134
Add data shapes 135
Simulate the process map in Arena 135
Check the drawing 135
Transfer the drawing to Arena 136
Simulate other Visio drawings 137
Use Visio and Arena in your organization 137
Distribute the Process Simulation add-in 137
Publish process maps on a Web page 138

Index **139**

Welcome

What is Arena software?

Arena software enables you to bring the power of modeling and simulation to your business. It is designed for analyzing the impact of changes involving significant and complex redesigns associated with supply chain, manufacturing, processes, logistics, distribution and warehousing, and service systems. Arena software provides the maximum flexibility and breadth of application coverage to model any desired level of detail and complexity.

Typical scenarios include:

- Detailed analysis of any type of manufacturing system, including material-handling components
- Analysis of complex customer service and customer management systems
- Analysis of global supply chains that include warehousing, transportation, and logistics systems
- Predicting system performance based on key metrics such as costs, throughput, cycle times, and utilizations
- Identifying process bottlenecks such as queue build ups and over-utilization of resources
- Planning staff, equipment, or material requirements.

Rockwell Software offers a full suite of products to provide enterprise-wide simulation, templates for various vertical markets, and an optimization template.

Intended audience

Arena software is designed for manufacturing or business process consultants and analysts and industrial or systems engineers. It is typically deployed as an enterprise business analysis and productivity tool.

We assume that you are familiar with the basic concepts and terms used in these types of systems. You are interested in improving business or manufacturing productivity and are responsible for evaluating and predicting the impact of proposed strategic and tactical changes to help improve performance. A familiarity with computers and the Microsoft® Windows® operating system is assumed. A familiarity with the concepts and terms used in simulation is also helpful.

Where can I go for help?

Our commitment to your success starts with the suite of learning aids and assistance we provide for Arena. Whether you're new to simulation or a seasoned veteran putting a new tool to use, you'll quickly feel at home with the Arena software.

Reference the user's guides

The documentation set for this package is comprised of user manuals for the software and template components, as well as a separate *Variables Guide* reference that provides complete descriptions of Arena variables found in the Arena product templates.

DOCUMENT CONVENTIONS

Throughout the guides, a number of style conventions are used to help identify material. New terms and concepts may be emphasized by use of italics or bold; file menu paths are in bold with a (>) separating the entries (e.g., go to **Help > Arena Help**); text you are asked to type is shown in Courier Bold (e.g., in this field, type **`Work Week`**), and dialog box and window button names are shown in bold (e.g., click **OK**).

Explore our examples

Arena is accompanied by a number of sample models that illustrate many of the commonly used approaches for capturing the essence of manufacturing processes. Examples are provided for both job shop and flow shop environments. For a description of and list of Arena's examples, go to **Help > Arena Help**. On the **Contents** tab, choose **Model Building Basics**, and then select **Viewing Arena Example Models**.

Get help

Online help is always at your fingertips! Arena incorporates the latest in help features, including *What's This?* help that displays a brief description of fields in dialogs, context-sensitive help on menu and toolbar buttons, and a help button on each of Arena's modules. Just refer to the Arena help table of contents and index for a list of all help topics.

Use the SMARTs library

As you craft models of your own manufacturing processes, use our SMARTs library to explore how to best use Arena. This suite of tutorial models covers topics ranging from modeling resources to animation techniques. The library is organized into categories to help you find the right model with ease. When you're wondering how to take the next step in your model, browse the SMARTs library for a ready-made solution. For a list of categories and their related SMARTS, go to **Help > Arena Help**. On the **Contents** tab, first click **Model Building Basics**, and then **Learning Arena with SMART Files**.

Access the Arena Symbol Factory

Arena animations can be enhanced using Arena Symbol Factory's extensive library of symbols. These symbols can be used for entity, resource, transporter or global pictures; or as graphic symbols within a model window. You can copy these symbols directly to the Arena model window, add them to your own libraries (*.plb* files), or add them to any of the Arena picture library files.

Get phone support

Rockwell Software provides full support for the entire Arena family of products. Questions concerning installation, how modules work, the use of the model editor, and the use of the software are handled by technical support.

ARENA TECHNICAL SUPPORT INCLUDES:

- (for users on active maintenance) a technical support hotline and e-mail address staffed by full-time, experienced professionals
- help with installation problems or questions related to the software's requirements
- troubleshooting
- limited support regarding the interaction of Arena with other programs
- support of the Arena Object Model, which is used in Microsoft Visual Basic for Applications

If you call the support line (1.440.646.5800), you should be at your computer and be prepared to give the following information:

- the product serial number
- the product version number
- the operating system you are using
- the exact wording of any messages that appeared on your screen
- a description of what happened and what you were doing when the problem occurred
- a description of how you tried to solve the problem

Get Web support

In addition to phone support, the Rockwell Automation Customer Support Center offers extensive online knowledgebases of tech notes and frequently asked questions for support of non-urgent issues. These databases are updated daily by our support specialists.

To receive regular e-mail messages with links to the latest tech notes, software updates, and firmware updates for the products that are of interest to you or to submit an online support request, go to support.rockwellautomation.com/supportrequests.

And be sure to check the Arena User Zone section of our Web site at www.ArenaSimulation.com. The User Zone links to a peer-to-peer forum on Arena topics and has a link to a

download page where you can check for possible software updates (patches). If you can't find the answer you need, contact your local representative or Arena technical support.

Get training

Do you need training? Rockwell Software offers a standard training course comprised of lecture and hands-on workshops designed to introduce you to the fundamental concepts of modeling with Arena.

We also offer customized training courses designed to meet your specific needs. These courses can be held in our offices or yours, and we can accommodate one person or twenty. You design the course that's right for you! Simply contact our consulting services group to discuss how we can help you achieve success in your simulation efforts.

Get consulting services

Rockwell Automation provides expert consulting and turnkey implementation of the entire Arena product suite. Please contact our offices for more information.

Contact us

We strive to help all of our customers become successful in their manufacturing improvement efforts. Toward this objective, we invite you to contact your local representative or Rockwell Software at any time that we may be of service to you.

Support E-mail: Arena-Support@software.rockwell.com
Corporate E-mail: Arena-Info@software.rockwell.com
Support phone: 1.440.646.5800
URL: www.ArenaSimulation.com
URL: www.rockwellsoftware.com

2 Getting Started

Introduction

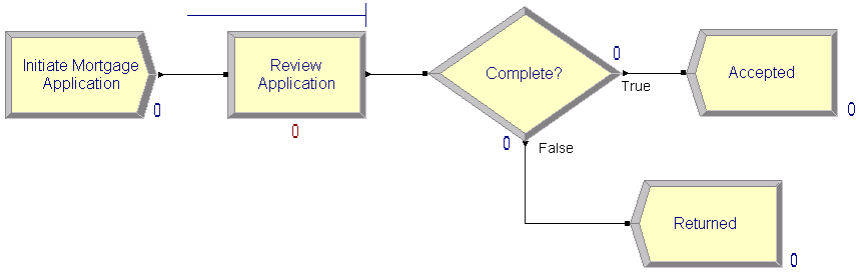
While you may not realize it quite yet, you now have the power to transform your business. Whenever you and others in your organization are wondering "what if…?," you can look into the future to find the answer.

With Arena, you can:

- **Model** your processes to define, document, and communicate.
- **Simulate** the future performance of your system to understand complex relationships and identify opportunities for improvement.
- **Visualize** your operations with dynamic animation graphics.
- **Analyze** how your system will perform in its "as-is" configuration and under a myriad of possible "to-be" alternatives so that you can confidently choose the best way to run your business.

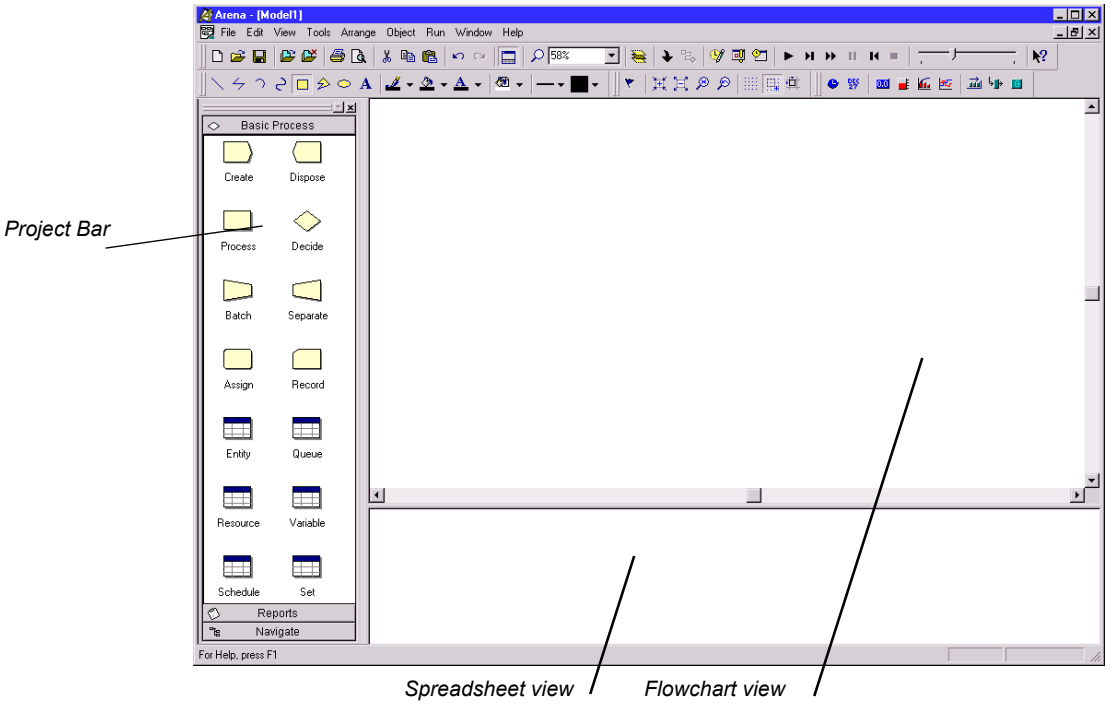
Our task: Analyze a home mortgage application process

In this chapter, we will examine a simple mortgage application process to illustrate how you can model, simulate, visualize, and analyze with Arena. To begin, we'll look at the process of receiving and reviewing a home mortgage application. We will build the flowchart shown below, introducing you to the process of modeling and simulating with Arena.



The Arena modeling environment

If Arena is not already running, start it from the Windows Start menu and navigate to **Programs > Rockwell Software > Arena**. The Arena modeling environment will open with a new model window, as shown below.



The Project Bar usually is docked at the left of the Arena application window, but can tear off or dock at another position like any toolbar.

To model your process in Arena, you'll work in three main regions of the application window. The *Project Bar* hosts panels with the primary types of objects that you will work with:

- **Basic Process**, **Advanced Process**, and **Advanced Transfer panels**: Contain the modeling shapes, called *modules*, that you'll use to define your process.
- **Reports panel**: Contains the reports that are available for displaying results of simulation runs.
- **Navigate panel**: Allows you to display different views of your model, including navigating through hierarchical submodels.

In the model window, there are two main regions. The *flowchart view* will contain all of your model graphics, including the process flowchart, animation, and other drawing elements. The lower, *spreadsheet view* displays model data, such as times, costs, and other parameters.

As we model the mortgage application process, we'll work in all three of these regions of Arena.

Map your process in a flowchart

Let's start by examining what we're going to do: **Build a flowchart**. The word itself—*flowchart*—suggests two of the main concepts behind modeling and simulation. We'll be building a *chart*—also referred to as a *process map* or a *model*—that describes a *flow.*

This raises a key question in process modeling: **What exactly is it that will *flow* through the chart?**

We're modeling the process of reviewing mortgage applications. These mortgage applications are the items, referred to as *entities,* that will move through the process steps in our model. They are the data, whether on paper or in electronic form, that are associated with our client's request for a mortgage. As we build the flowchart, it's helpful to think of the process from the perspective of the *entity* (the mortgage application), asking questions like:

- Where do the mortgage applications enter the process?
- What happens to them at each step?
- What resources are needed to complete work?

First, we'll draw the flowchart representing the mortgage application process. Refer to the Mortgage Application Process Flowchart (shown previously) so you'll know what we'll be creating.

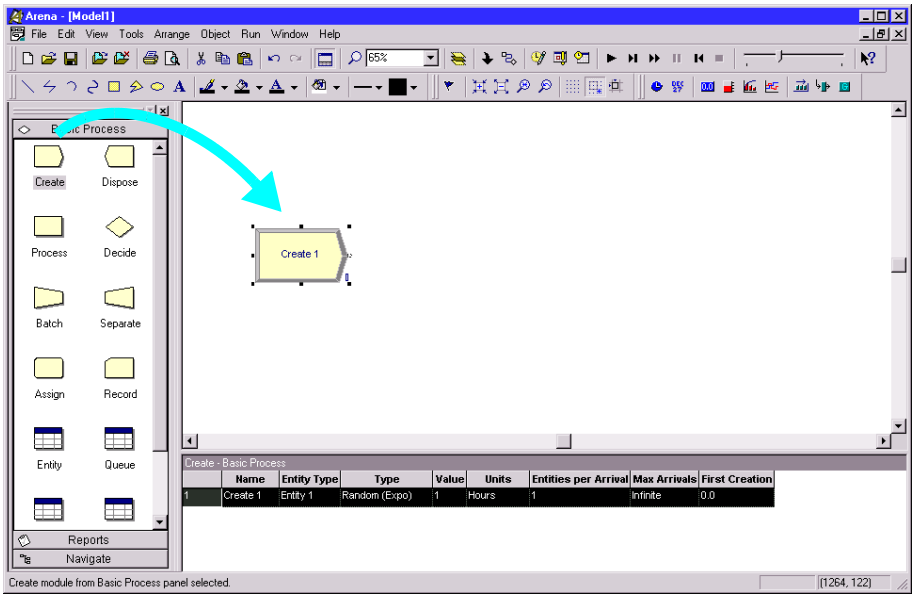
Create the mortgage application entities

We'll start the flowchart using a *Create module*, from the Basic Process panel. This is the starting point for the flow of entities through the model.

1. Drag the Create module from the Basic Process panel into the model window.

   A default name, Create 1, is given to the module when it's placed. We'll return later to provide a more meaningful description as well as some data to support the simulation.

Every process flow starts with a Create module. When you simulate the flowchart, individual entities will be created according to timing information you supply in the Create module properties. After it's created, each entity moves from the Create module to the next shape in the process flow.



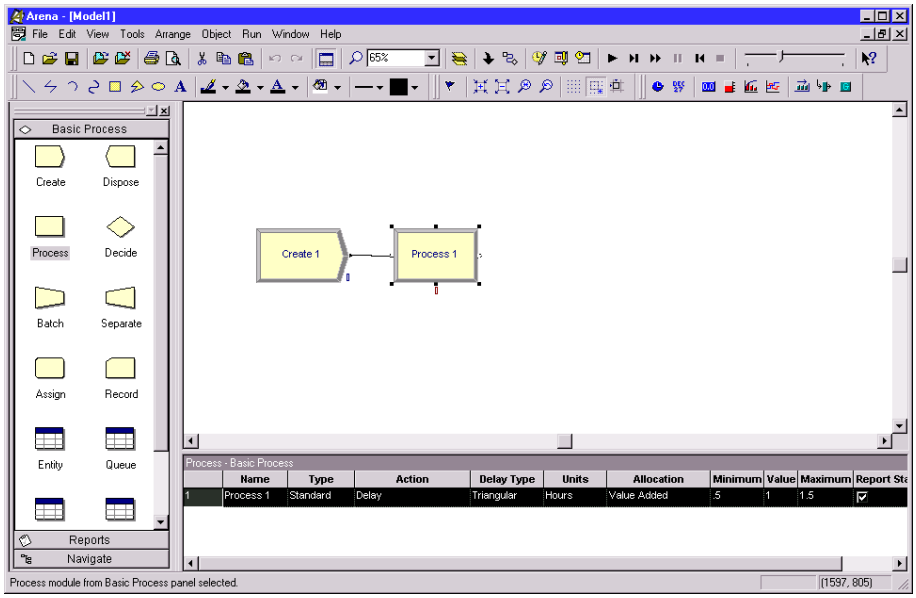
Process the applications

If your Create and Process weren't connected automatically when you placed the Process, check the Object > Auto-Connect menu to verify that it's checked. If it's not, select it to turn on this option.

Next in our flowchart is a *Process module*, from the Basic Process panel, representing the Review Application step.

1. So that Arena will automatically connect the Process to the Create module, be sure that the Create module is selected.
2. Drag a Process module from the Basic Process panel into the model window, placing it to the right of the Create. Arena will automatically connect the two modules.

As with the Create, the Process module has a default name that we'll replace later.



*Note: If no connection appears between Create and Process, click the **Object > Connect** menu item or the **Connect** toolbar button to draw a connection. Your cursor will change to a cross hair. Start the connection by clicking the exit point (▶) of the Create module, then click the entry point (■) of the Process module to complete the connection.*

How do I use Snap and Grid?

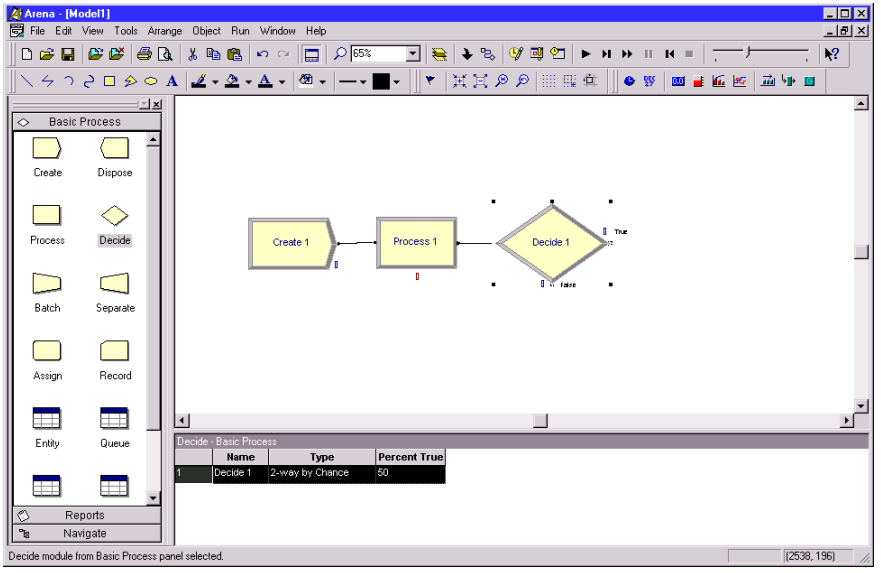
If your flowchart shapes aren't lining up properly, you can use Arena's snap and grid features to straighten them out. First, check the **Snap** option on the **View** menu so that newly placed shapes will be positioned at regular snap points. To realign the shapes you've already placed, select the main module shapes (the yellow boxes) by holding the **Ctrl** key and clicking on each shape. Then, click the **Arrange > Snap to Grid** menu option to adjust their positions to align with grid points.

You can display the grid by checking the **Grid** option on the **View** menu. Both snap and grid are turned off by clicking on the menu option again, turning off the check box.

Decide whether applications are complete

After the Process, we have a *Decide module*, from the Basic Process panel, which determines whether the mortgage application is complete.

1. If you're using the Auto-Connect feature (i.e., it's checked on the **Object > Auto-Connect** menu), be sure that the Process module is selected so that the Decide will be connected to it.
2. Drag a Decide module to the right of the Process module.



If the mortgage application has a complete set of information, it will leave the Decide module from the right side of the diamond shape, representing the True condition. Incomplete applications (False result to the Decide test) will leave via the bottom connection.

Dispose the applications to terminate the process

Next we'll place the *Dispose module*, from the Basic Process panel, representing accepted applications, connecting to the True (right) output from the Decide shape. Then, we'll complete the flowchart with another Dispose for returned applications.

1. Select the Decide shape so that our first Dispose will be connected automatically.

Entity flow always begins with a Create module and terminates with a Dispose module. You may have as many of each of these modules as you need to generate entities into the model and to remove them when their processing is complete.

2. Drag a Dispose module to the right of the Decide module. Arena will connect it to the primary (True) exit point of the Decide module. (We won't include a graphic display since you're now familiar with the drag-and-drop sequence.)
3. To add the second Dispose module, once again select the Decide module, so that Arena will automatically connect its False exit point to the new Dispose module, and drag another Dispose module below and to the right of the Decide module.
4. Drag and drop another Dispose module, placing it below and to the right of the Decide shape, completing the process flowchart.

What is a module?

In Arena, *modules* are the flowchart and data objects that define the process to be simulated. All information required to simulate a process is stored in modules.

For now, we're working with flowchart modules—those that are placed in the model window to describe the process. In the Basic Process panel, these are the first eight shapes:

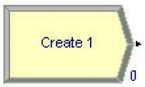
- **Create:** The start of process flow. Entities enter the simulation here.
- **Dispose:** The end of process flow. Entities are removed from the simulation here.
- **Process:** An activity, usually performed by one or more resources and requiring some time to complete.
- **Decide:** A branch in process flow. Only one branch is taken.
- **Batch:** Collect a number of entities before they can continue processing.
- **Separate:** Duplicate entities for concurrent or parallel processing, or separating a previously established batch of entities.
- **Assign:** Change the value of some parameter (during the simulation), such as the entity's type or a model variable.
- **Record:** Collect a statistic, such as an entity count or cycle time.

Simulation settings are defined in the **Run > Setup > Replication Parameters** dialog box. There is also a set of *data modules* for defining the characteristics of various process elements, such as resources and queues.

Define model data

Now that we've drawn the basic flowchart for our mortgage application process, let's define the data associated with the modules, including the name of the module and information that will be used when we simulate the process.

Initiate mortgage application (Create module)

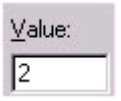


First, let's visit the Create module, which will be named *Initiate Mortgage Application*. Its data will include the type of entity to be created—in our case, a mortgage *Application*. We also need to define how often mortgage applications are initiated. We'll use an average of *2 hours* between applicants as a starting point for our model, and we'll make this a random activity to represent the natural variation in the timing of mortgage applications being submitted.

1. Double-click the Create module to open its property dialog box.
2. In the Name field, type **`Initiate Mortgage Application`**.

3. For the Entity Type, name our entities by typing **`Application`**.

4. Type **`2`** in the Value field of the Time Between Arrivals section.



5. Click **OK** to close the dialog box.

For now, we'll leave the default values for the other Create module properties. Feel free to explore their purposes through online help or the *Entity Arrivals* models in the SMARTs library.

What are entities?

Entities are the items—customers, documents, parts—that are being served, produced, or otherwise acted on by your process. In business processes, they often are documents or electronic records (checks, contracts, applications, purchase orders). In service systems, entities usually are people (the customers being served in a restaurant, hospital, airport, etc.). Manufacturing models typically have some kind of part running through the process, whether it's raw material, a subcomponent, or finished product. Other models might have different types of entities, such as data packets in network analysis or letters and boxes in package-handling facilities.

You may have different types of entities in the same model. For example, customers moving through a check-in counter at an airport might be separated into regular, first-class, and priority entity types. In some cases, entity types might be of an altogether different form rather than classifications of some basic type. For instance, in a pharmacy, prescriptions would be modeled as entities, running through the process of being filled. At the same time, customers might be competing for the pharmacist's attention with medical inquiries; they would also be modeled as entities.

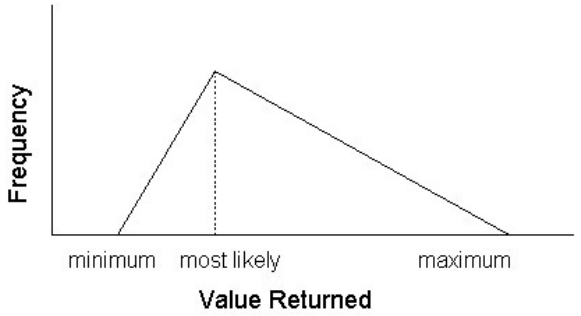
Review application (Process module)

Remember that as we create the flowchart, we're looking at the process from the perspective of the entity. The Create module is a starting point for an entity's flow through the system being modeled. Next, in our case, the application will be reviewed for completeness by a *Mortgage Review Clerk*. Because this will take some amount of time, holding the entity at this point in the flowchart for a *delay* and requiring a *resource* to perform the activity, we use a Process module. We'll call this process *Review Application*.

Appendix A describes the distributions available in Arena.

For the time delay, we also want to capture the natural variability that exists in most processes. Very often, for work done by people or equipment, a triangular distribution provides a good approximation. You specify the *minimum* time in which the work could be done, the *most likely value* for the time delay, and the *maximum* duration of the process.

During the simulation run, each time an entity enters the process, Arena will calculate a sample from the distribution information you've provided—in our case, a triangular distribution. Over the course of a long simulation run where thousands of individual samples are taken, the times will follow the profile illustrated next.



For our Review Application process, we'll use a minimum time of *1 hour*, most likely value of *1.75 hours*, and a maximum of *3 hours*. We will assign a *resource*, the *Mortgage Review Clerk*, to perform this process.

If more than one resource is required for a process to be performed, add as many as are necessary in the Process dialog's Resources list. An entity won't commence its process delay until all listed resources are available.

1. Double-click the Process module to open its property dialog box.
2. In the Name field, type **`Review Application`**.

3. To define a resource to perform this process, pull down the **Action** list and select **Seize Delay Release**.

   Arriving entities will wait their turn for the resource to be available. When its turn comes, the entity will *seize* the resource, *delay* for the process time, and then *release* the resource to do other work.

4. A list of resources will appear in the center of the dialog box. To add a resource for this process, click **Add**.

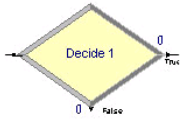
5. In the Resource Name field of the Resource dialog box, type **`Mortgage Review Clerk`**.

6. Click **OK** to close the Resource dialog box.
7. Define the process delay parameters in the Minimum, Most Likely Value, and Maximum fields as **`1`**, **`1.75`**, and **`3`**. (Note that the default delay type is Triangular and the default time units are in hours.)

8. Click **OK** to close the dialog box.

   For now, we'll leave the default values for the other Process module properties. Feel free to explore their purposes through online help or the "Modeling Concepts" and "Resources" models in the SMARTs library.

Complete? (Decide module)

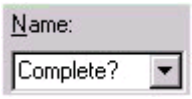


When you use a 2-way Decide module, the entity that enters the module leaves via one of the two exit points. If you want to make copies of an entity to model parallel processes, use a Separate module.

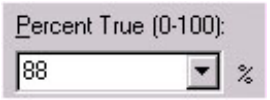
After the mortgage application has been reviewed, we determine whether to accept or return the application. In Arena, whenever an entity selects among branches in the process logic, taking just one of the alternatives, a Decide module is used.

For the mortgage application process, we'll use a simple probability to determine the outcome of the decision, with 88% of applications accepted as complete.

1. Double-click the Decide module to open its property dialog box.
2. In the Name field, type **`Complete?`**.



3. For the Percent True field, type **88** to define the percent of entities that will be treated with a "True" decision (i.e., will depart through the exit point at the right of the Decide module).



4. Click **OK** to close the dialog box.

Accepted, Returned (Dispose module)

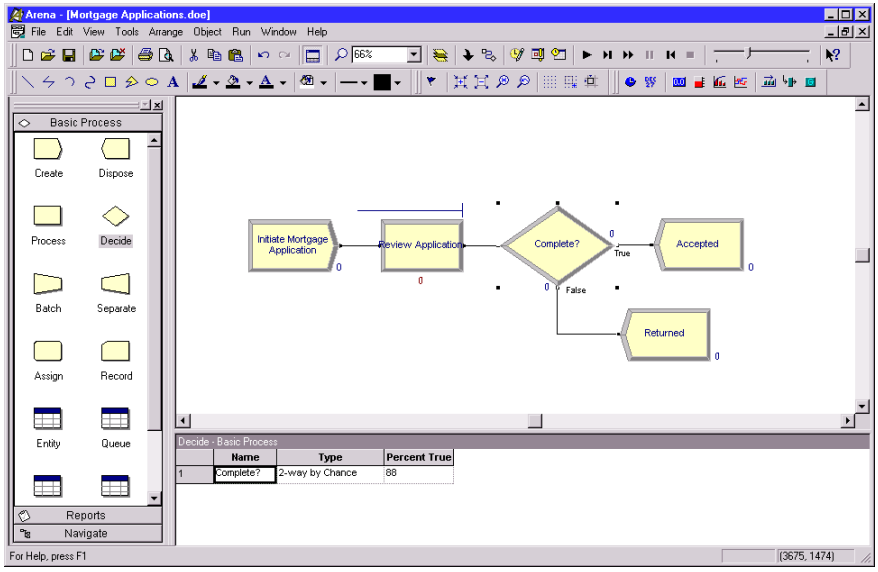
In our simple process for reviewing mortgage applications, all the work that we're interested in is done. Now, we'll remove the mortgage applications from the model, terminating the process with a Dispose module. Because there are two possible outcomes of the mortgage application process—applications can be accepted or returned—we're using two Dispose modules that will count the number of applications under each outcome.

1. Double-click the first Dispose module (connected to the True condition branch of the Decide module) to open its property dialog box, and in the Name field, type **Accepted**.

   Click **OK** to close the dialog box.

2. Double-click the other Dispose module to open its property dialog box. In the Name field, type **Returned**.

3. Click **OK** to close the dialog box.



Mortgage review clerk (Resource module)

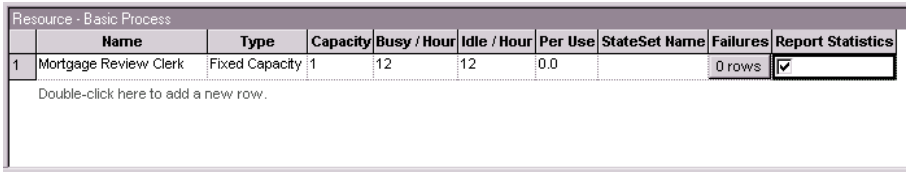
Along with our flowchart, we also can define parameters associated with other elements of our model, such as resources, entities, queues, etc. For the mortgage process, we'll simply define the cost rate for the Mortgage Review Clerk so that our simulation results will report the cost associated with performing this process. The clerk's costs are fixed at $12 per hour.

To provide these parameters to the Arena model, you'll enter them in the Resources spreadsheet.

You can edit the fields for any module using Arena's spreadsheet, including flowchart modules. Just click on the icon in the Basic Process panel to display its spreadsheet.

1. In the Basic Process panel, click the Resource icon to display the Resources spreadsheet.

2. Because we defined the Mortgage Review Clerk as the resource in the Review Application process, Arena has automatically added a resource with this name in the Resources spreadsheet. Click in the Busy/Hour cell and define the cost rate when the clerk is busy by typing **12**. Click in the Idle/Hour cell and assign the idle cost rate by typing **12**.



Prepare for the simulation

To make the model ready for simulation, we'll specify general project information and the duration of the simulation run. Since we're just testing our first-cut model, we'll perform a short, 20-day run.

1. Open the Project Parameters dialog box by using the **Run > Setup** menu item and clicking the **Project Parameters** tab. In the Project Title field, type **`Mortgage Review Analysis`**; we'll leave the Statistics Collection check boxes as the defaults, with Entities, Queues, Resources, and Processes checked and also check the costing box.
2. Next, click the **Replication Parameters** tab within the same Run Setup dialog box. In the Replication Length field, type **`20`**; and in the Time Units field directly to the right of Replication Length, select **days** from the drop-down list. Click **OK** to close the dialog box.

Save the simulation model

Now that you've done some work on your model, it seems like a good time to save it. Click **Save** on the Standard toolbar or select the **File > Save** menu item. Arena will prompt you for a destination folder and file name. Browse to the target folder in which you want to save the model (e.g., C:\My Documents) and type a name (e.g., **`Mortgage Review`**) in the file name field.

Arena's model files store all of the model definition, including the flowchart, other graphics you've drawn, and the module data you entered. When you perform a simulation run, the results are stored in a database using the same name as the model file.

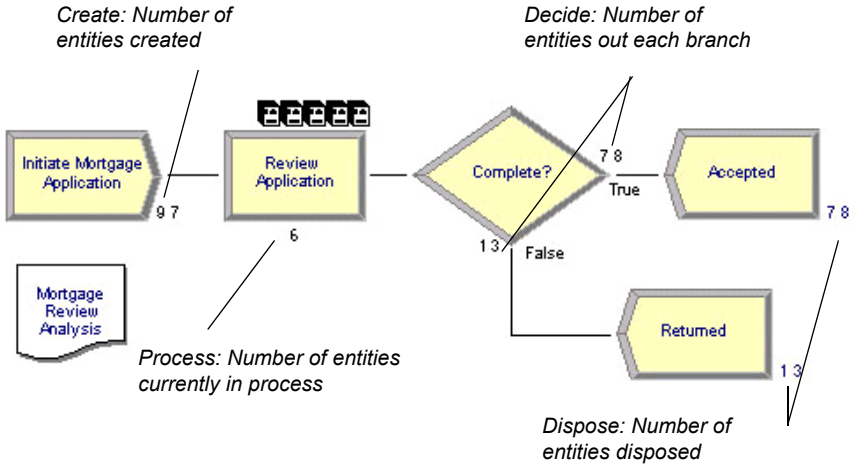
Simulate the process

With these few, short steps, we are ready to predict the future! The mortgage application model contains all of the information needed to run the simulation.

Start the simulation run by clicking the **Go** button or clicking the **Run > Go** menu item. Arena first will check to determine whether you've defined a valid model, then will launch the simulation.

*If Arena displays an error message, you can use the **Find** button in the error window to locate the source of the problem. You can change between the error and model windows by selecting them from the Window menu.*

As the simulation progresses, you'll see small entity pictures resembling pages moving among the flowchart shapes. Also, a variety of variables change values as entities are created and processed, as illustrated below.



The animation scale factor is the amount of simulated time between successive screen updates. Smaller values provide smoother, slower animation.

If the animation is moving too fast, you can slow it down by adjusting the animation scale factor. For this, you have two choices:

- Open the Run Setup dialog box via the **Run > Speed > Animation Speed Factor** menu item and enter a smaller value (e.g., **`0.005`**) for the scale factor; *or*
- Use the less-than (<) key during the run to decrease the scale factor by 20%. Be sure that the model window is active—not the Navigate panel—or > and < won't take effect. Pressing < repeatedly is an easy way to fine tune the animation speed. The greater-than (>) key speeds up animation by 20%.

*If the run finishes before you have a chance to explore these controls, answer No when you're asked if you want to view the results. Then click **Start Over** on the **Run** toolbar to begn the run again.*

To pause the simulation, click the **Pause** button or press the **Esc** key. With the automatic flowchart animation, you can see how many entities have been created, are currently in the Review Application process, have left each branch of our Decide module, and have left the model at each of our terminating Dispose modules. These variables can be helpful in verifying the model. For example, if the probability in the Decide shape was entered incorrectly (e.g., if you typed 12—the rejection probability—instead of 88), the variables would show that many more applications were leaving the Returned branch.

You also can step through the simulation one event at a time. **Pause** the simulation, then click the **Step** button or press the **F10** key. Each time you step the simulation, an entity is moved through the flowchart. Usually, you'll see animation of the entity's movement,

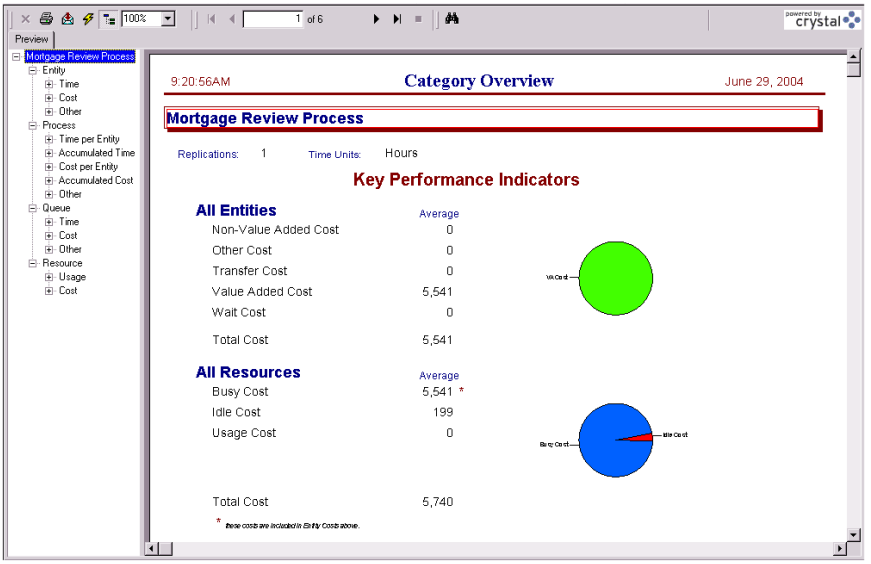
though sometimes no visual change will take place (e.g., when the next event is creating a new entity). When this occurs, just step again to move forward to the next event.

View simulation reports

After you've watched some of the animated flowchart, you can quickly run to the end of the simulation to view reports. **Pause** the simulation, then click the **Fast Forward** button to run the simulation without updating the animation.

At the end of the run, Arena will ask whether you'd like to view reports. Click **Yes**, and the default report (the Category Overview Report) will be displayed in a report window, as shown below.

Each of Arena's reports is displayed in its own window within the Arena application. You can use the standard window options (maximize, minimize, etc.) by clicking on the window control buttons or by pulling down the window menu.



On the left side of each report window is a tree listing the types of information available in the report. The project name (in our case, Mortgage Review) is listed at the top of the tree, followed by an entry for each category of data. This report summarizes the results across all replications (although, in this model, we have only one replication). Other reports provide detail for each replication.

By clicking on the entries inside the category sections, you can view various types of results from the simulation run. The following table illustrates some of the questions you could answer from the Category Overview Report on our simple mortgage application process simulation.

Question	Report Section	Answer
On average, how long did mortgage applications spend in the modeled process?	Total Time (Entity), Average column	16.51 hrs
What was the average cost of reviewing a mortgage application?	Total Cost (Entity), Average column	$22.99
What was the longest time an application spent in review?	Total Time (Process), Maximum column	33.45 hrs
What was the maximum number of applications waiting for review?	Number Waiting (Queue), Maximum column	21 applications
What proportion of time was the Mortgage Review Clerk busy?	Utilization (Resource), Average column	97%

After you've browsed the Category Overview Report, you can close it by clicking on the window icon to the left of the **File** menu and clicking **Close**. You can look at other reports by clicking on their icons in the Project Bar. Each report will be displayed in its own window. To return to the model window, close all of the report windows or select the model file from the Window menu.

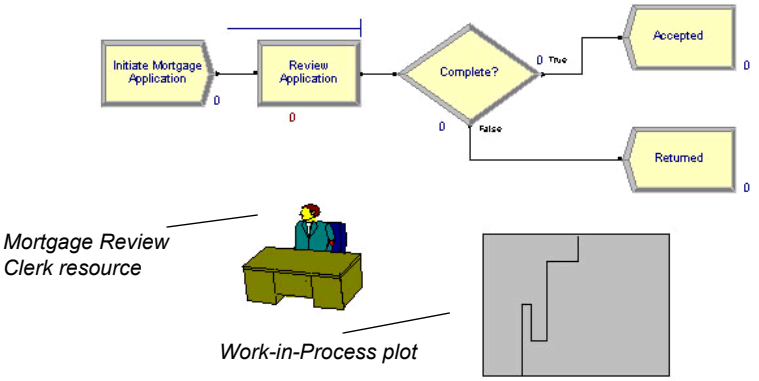
After you have viewed the reports and returned to the model window, end the Arena run session by clicking the **End** button.

Enhance the visualization process

Now that we've completed the basic steps for analyzing the mortgage application process, we can return to our model and embellish the graphical animation to gain further insight into the process dynamics. Animation also can be of great benefit in enticing others in the organization to be interested in process improvement.

We'll add two animation components to the mortgage model. First, we'll show our Mortgage Review Clerk working at a desk, either busy or idle. To gain a better sense of how many applications are waiting in the Review Application process over time, we'll also add a dynamic plot of the work-in-process (WIP) simulation variable. Our Arena model will appear as shown below after we add these two objects.

*Note: You can toggle between the split view (flowchart and spreadsheet) and a full-screen view of either area by clicking the **Split Screen** toolbar button or selecting the **View > Split Screen** menu item. When in full-screen view, clicking the icons on the Basic Process panel displays the appropriate view (flowchart for flowchart modules and spreadsheet for data-only modules).*



Animate the mortgage review clerk resource

During the simulation run, our Mortgage Review Clerk resource can be in one of two states. If no mortgage application entity is in-process, then the resource is *idle*. We'll use a picture of a person sitting at a desk to depict idleness. When an entity seizes the resource, the Mortgage Review Clerk's state is changed to *busy*, in which case our picture will show the person reviewing a document.

1. Click the **Resource** button on the **Animate** toolbar.
2. The Resource Placement dialog box appears. Select the **Mortgage Review Clerk** from the drop-down list in the Identifier field so that this object animates the Mortgage Review Clerk.
3. Open the Workers picture library by clicking the **Open** button, then browsing to the *Workers.plb* file in the Arena application folder (e.g., C:\Program Files\Rockwell Software\Arena) and double-clicking on it.
4. To change the idle picture:
   - Click the **Idle** button in the table on the left.
   - Select from the picture library table on the right the picture of the worker sitting down.
   - Click the **Transfer** button between the tables to use the worker picture for the Idle resource state.
5. To change the busy picture:
   - Click the **Busy** button in the table on the left.

- Select from the picture library table on the right the picture of the worker reading a document.
- Click the **Transfer** button between the tables to use the selected picture when the Mortgage Review Clerk is busy.

6. Click **OK** to close the dialog box. (All other fields can be left with their default values.)
7. The cursor will appear as a cross hair. Move it to the model window and click to place the Mortgage Review Clerk resource animation picture.
8. If you'd like to have the clerk appear a bit larger, select the picture and use the resize handles to enlarge it.

Plot the number of applications in-process

Our second animation enhancement is a plot of how many mortgage applications are under review as the simulation progresses. This will give us a sense of the dynamics of the workload, which can vary quite a bit when the random nature of processes is incorporated into a simulated model.

1. Click the **Plot** button on the **Animate** toolbar.
2. The Plot dialog box appears. We'll plot a single expression, the work-in-process (WIP) at the Review Application process. To add the expression, click **Add**.

You can plot many expressions on the same set of axes by adding multiple expressions in the Plot dialog. Each can be color-coded so that you can readily compare data such as workloads in processes, waiting customers, etc.

3. In the Plot Expression dialog box that appears, right-click in the Expression field to open the Expression Builder.
4. We want to plot the number of entities in our Review Application process over time. Select **Review Application** from the drop-down list in the Process Name field, then choose **WIP** from the Information drop-down list. Click **OK** to close the Expression Editor.
5. From our reports in the earlier simulation run, we noted that the maximum number of applications in the process was 9. Let's set our plot Maximum value to **`10`**.
6. In the History Points field, type **`5000`**, which will plot the most recent 5000 values of the variable during the simulation run. Click **OK** to close the Plot Expression dialog box. Note that Arena places the formula (Review Application.WIP) in the Plot Expression field.
7. To complete the plot definition, change the Time Range to **`480`**. Our plot's horizontal axis will represent 480 hours (20 days) of simulated time, matching our run length. Click **OK** to close the Plot dialog box.

8. The cursor changes to a cross hair. Draw the plot in the model window by clicking to locate each of the two opposite corners (e.g., the top-left and bottom-right corners), placing the plot below the flowchart and to the right of the resource.

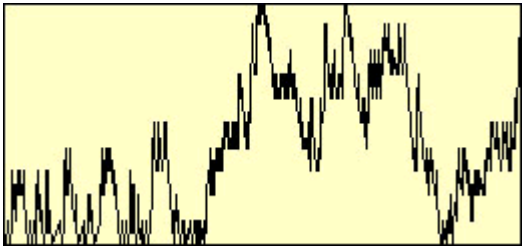
With the edits complete, you may want to save them by clicking **Save** or pressing **Ctrl+S**.

Rerun the simulation

Now that we've made our animation more interesting and valuable, let's run the simulation again. Because we didn't change any of the process parameters (i.e., data in the modules), the simulation will provide the same results.

Click **Run** (or press the **F5** key) to start the simulation. As the simulation progresses, you'll notice the Mortgage Review Clerk's picture change from idle (sitting at the desk) to busy (reading a document) and back again, as mortgage application entities move through the Review Application process.

The plot shows some significant peaks in the number of applications that are under review, caused by the combination of the variation in the time between arrivals of applications (defined in the Create module) and the time to process applications (Process module).



Next steps

You've succeeded in modeling, simulating, visualizing, and analyzing a simple mortgage application process. To further explore Arena's capabilities, try solving a few of these extensions to the process.

1. Add a screening process before the application is reviewed.

   Applications can be screened in as little as 15 minutes. Most often, it takes about 25 minutes for the screening, though sometimes it can require as much as 45 minutes. Assign a Receptionist (rate of $6.75/hour) to perform the screening. What proportion of the Receptionist's time will be used in this task?

2. Return some applications to the mortgage applicants after the screening process.

   On completion of the screening, 8% of the applications are returned. Also, because many of the deficient applications are caught in the new screening, the percentage of applications that are accepted in the formal review is raised from 88% to 94%, and the Mortgage Review process time is reduced by 10%. By how much did the cost of reviewing an application change? How about the total time to review applications?

To view completed Arena models for the main tutorial and these two extensions, browse to the Examples folder and open *Mortgage Applications.doe*, *Mortgage Extension 1.doe*, and *Mortgage Extension 2.doe*.

3 The Basic Process Panel

This chapter describes the flowchart and data modules that you'll use to model your process. Additionally, this chapter includes example uses of each module.

Flowchart modules

Flowchart modules are the set of objects that are placed in the model window to describe the simulation process.

Create module

DESCRIPTION

Number of entities created

This module is intended as the starting point for entities in a simulation model. Entities are created using a schedule or based on a time between arrivals. Entities then leave the module to begin processing through the system. The entity type is specified in this module.

TYPICAL USES

- The start of a part's production in a manufacturing line
- A document's arrival (e.g., order, check, application) into a business process
- A customer's arrival at a service process (e.g., retail store, restaurant, information desk)

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Entity Type	Name of the entity type to be generated.
Type	Type of arrival stream to be generated. Types include **Random** (uses an exponential distribution, user specifies mean), **Schedule** (uses an exponential distribution, mean determined from the specified Schedule module), **Constant** (user specifies constant value; e.g., 100), or **Expression** (drop-down list of various distributions).
Value	Determines the mean of the exponential distribution (if *Random* is used) or the constant value (if *Constant* is used) for the time between arrivals. Applies only when Type is Random or Constant.
Schedule Name	Identifies the name of the schedule to be used. The schedule defines the arrival pattern for entities arriving to the system. Applies only when Type is Schedule.

Prompt	Description
Expression	Any distribution or value specifying the time between arrivals. Applies only when Type is Expression.
Units	Time units used for interarrival and first creation times. Does not apply when Type is Schedule.
Entities per Arrival	Number of entities that will enter the system at a given time with each arrival.
Max Arrivals	Maximum number of entities that this module will generate. When this value is reached, the creation of new entities by this module ceases.
First Creation	Starting time for the first entity to arrive into the system. Does not apply when Type is Schedule.

Dispose module

DESCRIPTION

Number of entities that have been disposed

This module is intended as the ending point for entities in a simulation model. Entity statistics may be recorded before the entity is disposed.

TYPICAL USES

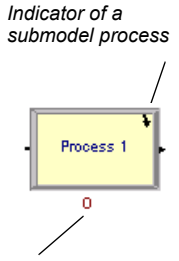
- Parts leaving the modeled facility
- The termination of a business process
- Customers departing the store

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Record Entity Statistics	Determines whether or not the incoming entity's statistics will be recorded. Statistics include value-added time, non-value-added time, wait time, transfer time, other time, total time, value-added cost, non-value-added cost, wait cost, transfer cost, other cost, and total cost.

Process module

DESCRIPTION



This module is intended as the main processing method in the simulation. Options for seizing and releasing resource constraints are available. Additionally, there is the option to use a "submodel" and specify hierarchical user-defined logic. The process time is allocated to the entity and may be considered to be value added, non-value added, transfer, wait, or other. The associated cost will be added to the appropriate category.

TYPICAL USES

- Machining a part
- Reviewing a document for completeness
- Fulfilling orders
- Serving a customer

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of specifying logic within the module. **Standard** processing signifies that all logic will be stored within the Process module and defined by a particular Action. **Submodel** indicates that the logic will be hierarchically defined in a "submodel" that can include any number of logic modules.
Action	Type of processing that will occur within the module. **Delay** simply indicates that a process delay will be incurred with no resource constraints. **Seize Delay** indicates that a resource(s) will be allocated in this module and delay will occur, but that resource release will occur at a later time. **Seize Delay Release** indicates that a resource(s) will be allocated followed by a process delay and then the allocated resource(s) will be released. **Delay Release** indicates that a resource(s) has previously been allocated and that the entity will simply delay and release the specified resource(s). Applies only when Type is Standard.
Priority	Priority value of the entity waiting at this module for the resource(s) specified if one or more entities are waiting for the same resource(s) anywhere in the model. Not visible when Action is Delay or Delay Release or when Type is Submodel.
Resources	Lists the resources or resource sets used for entity processing. Does not apply when Action is Delay, or when Type is Submodel.

Prompt	Description
Delay Type	Type of distribution or method of specifying the delay parameters. **Constant** and **Expression** require single values, while **Normal**, **Uniform**, and **Triangular** require several parameters.
Units	Time units for delay parameters.
Allocation	Determines how the processing time and process costs will be allocated to the entity. The process may be considered to be **Value Added**, **Non-Value Added**, **Transfer**, **Wait**, or **Other** and the associated cost will be added to the appropriate category for the entity and process.
Minimum	Parameter field for specifying the minimum value for either a uniform or triangular distribution.
Value	Parameter field for specifying the **mean** for a normal distribution, the **value** for a constant time delay, or the **mode** for a triangular distribution.
Maximum	Parameter field for specifying the maximum value for either a uniform or triangular distribution.
Std Dev	Parameter field for specifying the standard deviation for a normal distribution.
Expression	Parameter field for specifying an expression whose value is evaluated and used for the processing time delay.
Report Statistics	Specifies whether or not statistics will be automatically collected and stored in the report database for this process.

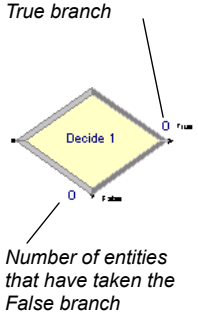
Process module — Resource dialog

PROMPTS

Prompt	Description
Type	Specification of a particular resource, or selecting from a pool of resources (i.e., a resource set).
Resource Name	Name of the resource that will be seized and/or released. Applies only when Type is Resource.
Set Name	Name of the resource set from which a member will be seized and/or released. Applies only when Type is Set
Quantity	Number of resources of a given name or from a given set that will be seized/released. For sets, this value specifies only the number of a selected resource that will be seized/released (based on the resource's capacity), not the number of members of a set to be seized/released.
Selection Rule	Method of selecting among available resources in a set. Cyclical will cycle through available members (e.g., 1st member–2nd member–3rd member–1st member–2nd member–3rd member). **Random** will randomly select a member. **Preferred Order** will always select the first available member (1st member, if available; then 2nd member, if available; then 3rd member, etc.). **Specific Member** requires an input attribute value to specify which member of the set (previously saved in the Save Attribute field). **Largest Remaining Capacity** and **Smallest Number Busy** are used for resources with multiple capacity. Applies only when Type is Set.
Save Attribute	Attribute name used to save the index number into the set of the member that is selected. This attribute can later be referenced with the Specific Member selection rule. Does not apply when Selection Rule is Specific Member. If Action is specified as Delay Release, the value specified defines which member (the index number) of the set to be released. If no attribute is specified, the entity will release the member of the set that was *last* seized.
Set Index	The index number into the set of the member requested. Applies only when Selection Rule is Specific Member. If Action is specified as Delay Release, the value specified defines which member (the index number) of the set is to be released.

Decide module

DESCRIPTION



This module allows for decision-making processes in the system. It includes options to make decisions based on one or more conditions (e.g., if entity type is Gold Card) or based on one or more probabilities (e.g., 75%, true; 25%, false). Conditions can be based on attribute values (e.g., Priority), variable values (e.g., Number Denied), the entity type, or an expression (e.g., NQ(ProcessA.Queue)).

There are two exit points out of the Decide module when its specified type is either 2-way by Chance or 2-way by Condition. There is one exit point for "true" entities and one for "false" entities. When the N-way by Chance or by Condition type is specified, multiple exit points are shown for each condition or probability and a single "else" exit. The number of entities that exit from each type (true/false) is displayed for 2-way by Chance or by Condition modules only.

TYPICAL USES

- Dispatching a faulty part for rework
- Branching accepted vs. rejected checks
- Sending priority customers to a dedicated process

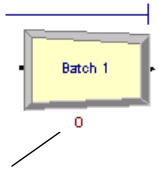
PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Indicates whether the decision is based on a condition (if X>Y) or by chance/percentage (e.g., 60%, yes; 40%, no). The type can be specified as either 2-way or N-way. 2-way allows for one condition or probability (plus the "false" exit). N-way allows for any number of conditions or probabilities to be specified as well as an "else" exit.
Conditions	Defines one or more conditions used to direct entities to different modules. Applies only when Type is N-way by Condition.
Percentages	Defines one or more percentages used to direct entities to different modules. Applies only when Type is N-way by Chance.
Percent True	Value that will be checked to determine the percentage of entities sent out a given True exit.
If	Types of conditions that are available for evaluation: Variable, Variable Array (1D), Variable Array (2D), Attribute, Entity Type, Expression.

Prompt	Description
Named	Specifies either the name of the variable, attribute, orentity type that will be evaluated when an entity enters the module. Does not apply when Type is Expression.
Is	Evaluator for the condition. Applies only to Attribute and Variable conditions.
Row	Specifies the row index for a variable array. Applies only when Type is N-way by Condition or 2-way by Condition and Variable is Array 1-D or Array 2-D.
Column	Specifies the column index for a variable array. Applies only when Type is N-way by Condition or 2-way by Condition and Variable is Array 1-D or Array 2-D.
Value	Expression that will be either compared to an attribute or variable or that will be evaluated as a single expression to determine if it is true or false. Does not apply to Entity Type condition. If Type is Expression, this value must also include the evaluator (e.g., Color<>Red).

Batch module

DESCRIPTION



Number of entities waiting to be batched

This module is intended as the grouping mechanism within the simulation model. Batches can be permanently or temporarily grouped. Temporary batches must later be split using the Separate module.

Batches may be made with any specified number of entering entities or may be matched together based on an attribute. Entities arriving at the Batch module are placed in a queue until the required number of entities has accumulated. Once accumulated, a new representative entity is created.

TYPICAL USES

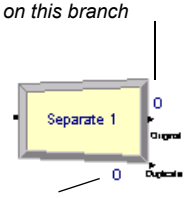
- Collect a number of parts before starting processing
- Reassemble previously separated copies of a form
- Bring together a patient and his record before commencing an appointment

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of batching entities together.
Batch Size	Number of entities to be batched.
Save Criterion	Method for assigning representative entity's user-defined attribute values.
Rule	Determines how incoming entities will be batched. **Any Entity** will take the first "Batch Size" number of entities and put them together. **By Attribute** signifies that the values of the specified attribute must match for entities to be grouped. For example, if Attribute Name is Color, all entities must have the same Color value to be grouped; otherwise, they will wait at the module for additional incoming entities.
Attribute Name	Name of the attribute whose value must match the value of the other incoming entities in order for a group to be made. Applies only when Rule is By Attribute.

Separate module

DESCRIPTION



This module can be used to either copy an incoming entity into multiple entities or to split a previously batched entity. Rules for allocating costs and times to the duplicate are also specified. Rules for attribute assignment to member entities are specified as well.

When splitting existing batches, the temporary representative entity that was formed is disposed and the original entities that formed the group are recovered. The entities proceed sequentially from the module in the same order in which they originally were added to the batch.

When duplicating entities, the specified number of copies is made and sent from the module. The original incoming entity also leaves the module.

TYPICAL USES

- Send individual entities to represent boxes removed from a container
- Send an order both to fulfillment and billing for parallel processing
- Separate a previously batched set of documents

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of separating the incoming entity. **Duplicate Original** will simply take the original entity and make some number of identical duplicates. **Split Existing Batch** requires that the incoming entity be a temporarily batched entity using the Batch module. The original entities from the batch will be split.
Percent Cost to Duplicates	Allocation of costs and times of the incoming entity to the outgoing duplicates. This value is specified as a percentage of the original entity's costs and times (between 0-100). The percentage specified will be split evenly between the duplicates, while the original entity will retain any remaining cost/time percentage. Visible only when Type is Duplicate Original.
# of Duplicates	Number of outgoing entities that will leave the module, in addition to the original incoming entity. Applies only when Type is Duplicate Original.
Member Attributes	Method of determining how to assign the representative entity attribute values to the original entities.These options relate to six of the special-purpose attributes (Entity.Type, Entity.Picture, Entity.Sequence, Entity.Station, Entity.Jobstep, and Entity.HoldCostRate) and all user-defined attributes. Applies only when Type is Split Existing Batch.
Attribute Name	Name of representative entity attribute(s) that are assigned to original entities of the group. Applies only when Member Attributes is Take Specific Representative Values.

Assign module

DESCRIPTION

This module is used for assigning new values to variables, entity attributes, entity types, entity pictures, or other system variables. Multiple assignments can be made with a single Assign module.

TYPICAL USES

- Accumulate the number of subassemblies added to a part
- Change an entity's type to represent the customer copy of a multi-page form
- Establish a customer's priority

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Assignments	Specifies the one or more assignments that will be made when an entity executes the module.
Type	Type of assignment to be made. Other can include system variables, such as resource capacity or simulation end time.
Variable Name	Name of the variable that will be assigned a new value when an entity enters the module. Applies only when Type is Variable, Variable Array (1D), or Variable Array (2D).
Row	Specifies the row index for a variable array.
Column	Specifies the column index for a variable array.
Attribute Name	Name of the entity attribute that will be assigned a new value when the entity enters the module. Applies only when Type is Attribute.
Entity Type	New entity type that will be assigned to the entity when the entity enters the module. Applies only when Type is Entity Type.
Entity Picture	New entity picture that will be assigned to the entity when the entity enters the module. Applies only when Type is Entity Picture.
Other	Identifies the special system variable that will be assigned a new value when an entity enters the module. Applies only when Type is Other.
New Value	Assignment value of the attribute, variable, or other system variable. Does not apply when Type is Entity Type or Entity Picture.

Record module

DESCRIPTION

This module is used to collect statistics in the simulation model. Various types of observational statistics are available, including time between exits through the module, entity statistics (time, costing, etc.), general observations, and interval statistics (from some time stamp to the current simulation time). A count type of statistic is available as well. Tally and Counter sets can also be specified.

TYPICAL USES

- Collect the number of jobs completed each hour
- Count how many orders have been late being fulfilled
- Record the time spent by priority customers in the main check-out line

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Type of observational (tally) or count statistic to be generated. **Count** will increase or decrease the value of the named statistic by the specified value. **Entity Statistics** will generate general entity statistics, such as time and costing/duration information. **Time Interval** will calculate and record the difference between a specified attribute's value and current simulation time. **Time Between** will track and record the time between entities entering the module. **Expression** will record the value of the specified expression.
Attribute Name	Name of the attribute whose value will be used for the interval statistics. Applies only when Type is Interval.
Value	Value that will be recorded to the observational statistic when Type is Expression or added to the counter when Type is Count.
Tally Name	This field defines the symbol name of the tally into which the observation is to be recorded. Applies only when Type is Time Interval, Time Between, or Expression.
Counter	This field defines the symbol name of the counter to *Name* increment/decrement. Applies only when Type is Counter.
Record into Set	Check box to specify whether or not a tally or counter set will be used.
Tally Set Name	Name of the tally set that will be used to record the observational-type statistic. Applies only when Type is Time Interval, Time Between, or Expression.
Counter Set Name	Name of the counter set that will be used to record the count-type statistic. Applies only when Type is Count.
Set Index	Index into the tally or counter set.

Data modules

Data modules are the set of objects in the spreadsheet view of the model that define the characteristics of various process elements, such as resources and queues.

Entity module

DESCRIPTION

This data module defines the various entity types and their initial picture values in a simulation. Initial costing information and holding costs are also defined for the entity.

TYPICAL USES

- Items being produced or assembled (parts, pallets)
- Documents (forms, e-mails, faxes, reports)
- People moving through a process (customers, callers)

PROMPTS

Prompt	Description
Entity Type	The name of the entity type being defined. This name must be unique.
Initial Picture	Graphical representation of the entity at the start of the simulation. This value can be changed during the simulation using the Assign module.
Holding Cost/Hour	Hourly cost of processing the entity through the system. This cost is incurred when the entity is anywhere in the system.
Initial VA Cost	Initial cost value that will be assigned to the value-added cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a value-added activity.
Initial NVA Cost	Initial cost value that will be assigned to the non-value-added cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a non-value-added activity.
Initial Waiting Cost	Initial cost value that will be assigned to the waiting-cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a wait activity; e.g., waiting to be batched or waiting for resource(s) at a Process module.
Initial Transfer Cost	Initial cost value that will be assigned to the transfer cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in a transfer activity.

Prompt	Description
Initial Other Cost	Initial cost value that will be assigned to the other cost attribute of the entity. This attribute accrues the costs incurred when an entity is spending time in another activity.
Report Statistics	Specifies whether or not statistics will be collected automatically and stored in the report database for this entity type.

Queue module

DESCRIPTION

This data module may be utilized to change the ranking rule for a specified queue. The default ranking rule for all queues is First In, First Out unless otherwise specified in this module. There is an additional field that allows the queue to be defined as shared.

TYPICAL USES

- Stack of work waiting for a resource at a Process module
- Holding area for documents waiting to be collated at a Batch module

PROMPTS

Prompt	Description
Name	The name of the queue whose characteristics are being defined. This name must be unique.
Type	Ranking rule for the queue, which can be based on an attribute. Types include **First In, First Out**; **Last In, First Out**; **Lowest Attribute Value** (first); and **Highest Attribute Value** (first). A low attribute value would be 0 or 1, while a high value may be 200 or 300.
Attribute Name	Attribute that will be evaluated for the Lowest Attribute Value or Highest Attribute Value types. Entities with lowest or highest values of the attribute will be ranked first in the queue, with ties being broken using the First In, First Out rule.
Shared	Check box that determines whether a specific queue is used in multiple places within the simulation model. Shared queues can only be used for seizing resources (e.g., with the Seize module from the Advanced Process panel).
Report Statistics	Specifies whether or not statistics will be collected automatically and stored in the report database for this queue.

Resource module

DESCRIPTION

This data module defines the resources in the simulation system, including costing information and resource availability. Resources may have a fixed capacity that does not vary over the simulation run or may operate based on a schedule. Resource failures and states can also be specified in this module.

TYPICAL USES

- Equipment (machinery, cash register, phone line)
- People (clerical, order processing, sales clerks, operators)

PROMPTS

Prompt	Description
Name	The name of the resource whose characteristics are being defined. This name must be unique.
Type	Method for determining the capacity for a resource. **Fixed Capacity** will not change during the simulation run. **Based on Schedule** signifies that a Schedule module is used to specify the capacity and duration information for the resource.
Capacity	Number of resource units of a given name that are available to the system for processing. Applies only when Type is Fixed Capacity.
Schedule Name	Identifies the name of the schedule to be used by the resource. The schedule defines the capacity of a resource for a given period of time. Applies only when type is Schedule.
Schedule Rule	Dictates when the actual capacity change is to occur when a decrease in capacity is required for a busy resource unit. Applies only when Type is Schedule.
Busy/Hour	Cost per hour of a resource that is processing an entity. The resource becomes busy when it is originally allocated to an entity and becomes idle when it is released. During the time when it is busy, cost will accumulate based on the busy/hour cost. The busy cost per hour is automatically converted to the appropriate base time unit specified within the Replication Parameters page of the **Run > Setup** menu item.
Idle/Hour	Cost per hour of a resource that is idle. The resource is idle while it is not processing an entity. During the time when it is idle, cost will accumulate based on the idle/hour cost. The idle cost per hour is automatically converted to the appropriate base time unit specified within the Replication Parameters page of the **Run > Setup** menu item.

Prompt	Description
Per Use	Cost of a resource on a usage basis, regardless of the time for which it is used. Each time the resource is allocated to an entity, it will incur a per-use cost.
StateSet Name	Name of states that the resource may be assigned during the simulation run.
Initial State	Initial state of a resource. If specified, the name must be defined within the repeat group of state names. This field is shown only when a StateSet Name is defined.
Failures	Lists all failures that will be associated with the resource. ■ *Failure Name*–Name of the failure associated with the resource. ■ *Failure Rule*–Behavior that should occur when a failure is to occur for a busy resource unit.
Report Statistics	Specifies whether or not statistics will be collected automatically and stored in the report database for this resource.

Variable module

DESCRIPTION

This data module is used to define a variable's dimension and initial value(s). Variables can be referenced in other modules (e.g., the Decide module), can be reassigned a new value with the Assign module, and can be used in any expression.

There are three methods for manually editing the Initial Values of a Variable module:

- Via the standard spreadsheet interface. In the module spreadsheet, right-click on the **Initial Values** cell and select the **Edit via spreadsheet** menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value.
- Via the module dialog box. In the module spreadsheet, right-click on any cell and select the **Edit via dialog** menu item. The values for two-dimensional arrays should be entered one column at a time. Array elements not explicitly assigned are assumed to have the last entered value.
- Via the two-dimensional (2-D) spreadsheet interface. In the module spreadsheet, click on the **Initial Values** cell.

TYPICAL USES

- Number of documents processed per hour
- Serial number to assign to parts for unique identification
- Space available in a facility

PROMPTS

Prompt	Description
Name	The unique name of the variable being defined.
Rows	Number of rows in a dimensional variable.
Columns	Number of columns in a dimensional variable.
Statistics	Check box for determining whether or not statistics will be collected. This field is visible when the rows and columns are not specified (single variables).
Clear Option	Defines the time (if at all) when the value(s) of the variable is reset to the initial value(s) specified. Specifying **Statistics** indicates to reset this variable to its initial value(s) whenever statistics are cleared. Specifying **System** indicates to reset this variable to its initial value(s) whenever the system is cleared. **None** indicates to never reset this variable to its initial value(s), except prior to the first replication.
Initial Values	Lists the initial value(s) of the variable. This value(s) may be changed with the Assign module.
Initial Value	Variable value at the start of the simulation.
Report Statistics	Specifies whether or not statistics will be collected automatically and stored in the report database for this variable.

Schedule module

DESCRIPTION

This data module may be used in conjunction with the Resource module to define an operating schedule for a resource or with the Create module to define an arrival schedule. Additionally, a schedule may be used and referenced to factor time delays based on the simulation time. Duration-formatted schedules are defined within this module. Calendar-formatted schedules are defined by selecting **Edit > Calendar Schedules > Time Patterns**. (For more details, refer to the section entitled "Calendar schedule information" on page 45.)

TYPICAL USES

- Work schedule for staff, including breaks
- Breakdown patterns for equipment
- Volume of customers arriving at a store
- Learning-curve factors for new workers

PROMPTS

Prompt	Description
Name	The name of the schedule being defined. This name must be unique.
Format Type	Format of schedule being defined. If format is duration, the schedule is defined with a collection of value duration pairs. If format is calendar, the schedule is defined using the Time Pattern editor.
Type	Type of schedule being defined. This may be **Capacity** related (for resource schedules), **Arrival** related (for the Create module), or **Other** (miscellaneous time delays or factors).
Time Units	Time units used for the time-duration information.
Scale Factor	Method of scaling the schedule for increases or decreases in Arrival/Other values. The specified Value fields will be multiplied by the scale factor to determine the new values. Not available for Capacity-type schedules.
Durations	Lists the value and duration pairs for the schedule. Values can be capacity, arrival, or other type values, while the duration is specified in time units. Schedule pairs will repeat after all durations have been completed, unless the last duration is left blank (infinite). Schedule data can be entered graphically using the graphical schedule editor or manually using the Value/Duration fields.
Value	Represents either the capacity of a resource (if Type is Capacity), arrival rate (if Type is Arrival), or some other value (if Type is Other). Examples of Other may be a factor that is used in a delay expression to scale a delay time during various parts of the day.
Duration	Time duration for which a specified Value will be valid.

Set module

DESCRIPTION

This data module defines various types of sets, including resource, counter, tally, entity type, and entity picture. Resource sets can be used in the Process modules (and Seize, Release, Enter, and Leave of the Advanced Process and Advanced Transfer panels). Counter and Tally sets can be used in the Record module. Queue sets can be used with the Seize, Hold, Access, Request, Leave, and Allocate modules of the Advanced Process and Advanced Transfer panels.

TYPICAL USES

- Machines that can perform the same operations in a manufacturing facility
- Supervisors, check-out clerks in a store
- Shipping clerks, receptionists in an office
- Set of pictures corresponding to a set of entity types

PROMPTS

Prompt	Description
Name	The unique name of the set being defined.
Type	Type of set being defined.
Members	Repeat group that specifies the resource members with the set. The order of listing the members within the repeat group is important when using selection rules such as Preferred Order and Cyclical.
Resource Name	Name of the resource to include in the resource set. Applies only when Type is Resource.
Tally Name	Name of the tally within the tally set. Applies only when Type is Tally.
Counter Name	Name of the counter within the counter set. Applies only when Type is Counter.
Entity Type	Name of the entity type within the entity type set. Applies only when Type is Entity.
Picture Name	Name of the picture within the picture set. Applies only when Type is Entity Picture.

Calendar schedule information

Time Pattern

DESCRIPTION

A Time Pattern defines a series of values that change over time. For example, a worker might be on duty during a standard work shift from Monday through Friday. The availability of the worker could be represented as a value of 1 when he's on duty, 0 when he's off duty. The time pattern that defines this availability schedule would list the changes from on duty to off duty over the week.

Time patterns are saved in an Arena model in the Schedule module with a Type set to Calendar. Their data are edited via the Time Pattern window, which is opened from the **Edit > Calendar Schedules > Time Patterns** menu option.

TYPICAL USES

- Weekly work schedules (e.g., 1-shift, 5-day weeks; 2-shift, 4-day weeks)
- Daily work patterns (e.g., 8-hour day with 30-minute lunch break)
- Arrival patterns (e.g., weekly customer arrivals, daily part orders)

PROMPTS

Prompt	Description
Category	The unique name of the category to which the time pattern belongs. A category named Uncategorized is provided by default.
Name	The unique name of this time pattern.
Time Pattern Type	The type of data to be defined in this time pattern. Types include Arrival, Capacity, Efficiency, and Other.
Default Value Type	The data type defining the default value for this time pattern. Types include Numeric, State, and Time Pattern.
Default Value	The value to be used for all time periods where no other value has been defined in this time pattern. The Default Value Type dictates the information supplied in this field: ■ *Numeric*—a non-negative, real number (e.g., 1) ■ *State*—the name of a State, defined via the States editor dialog box (e.g., On Shift) ■ *Time Pattern*—another time pattern of matching Time Pattern Type (e.g., Standard Work Day)

Prompt	Description
Duration	The length of this time pattern, which establishes the period in which the time pattern will repeat. For example, a time pattern of Duration equal to 1 day will repeat every 24 hours. This field is read-only, displaying the settings chosen in the Duration Edit dialog box for this time pattern.
Base Date	The beginning calendar date of this time pattern. Applies only when the time pattern type is Custom.
Scale Factor	A multiplier to be applied to all values in the time pattern and specified as a non-negative, real value.
Lock Times	Indicates whether the times defined in this time pattern are absolute clock times (when Lock Times is True), or are relative times from the beginning of the time pattern (when Lock Times is False). Time patterns with relative times are typically used as building blocks for other time patterns. For example, a shift time pattern (with Lock Times as False) might be created that establishes the timing of breaks relative to the start of the shift. A weekly pattern (with Lock Times as True) then could use that shift pattern repeatedly to establish the working schedule for first shift starting at 7:00, second shift starting at 15:00, etc.
Time Spans List	Values to apply for this time pattern and the time periods during which the values should apply. If any time spans overlap, those defined further down in the list supercede values defined above.
Value Type	The data type defining the value for this time span. Types include Numeric, State, and Time Pattern.
Value	The value to be used during this time span. Valid entries are as described for the Default Value.
Start	Starting time for the time span, defined as a day, hour, minute, second, and millisecond. Day values are specified as a day number (for custom durations) or a day-of-week (for 1-week durations). All times are specified in 24-hour format (e.g., 3:00 PM is 15 hours, 0 minutes).
End	Ending time for the time span, specified as described for the Start time.

Calendar Exceptions

DESCRIPTION

An Exception defines a value that is to be applied for a resource over some fixed period of time. The exception value overrides the base value defined for the resource. For example, if a machine's capacity was scheduled to be available (value of 1) during second shift according to its base time pattern, you could add an exception to change it to unavailable for second shift on a particular day (e.g., December 30, 2004).

Exception data are edited via the Exception and Composite windows, which are opened from the **Edit > Calendar Schedules > Exceptions** and **Display Composite** menu options.

TYPICAL USES

- Overtime
- Holidays
- Regularly scheduled meetings

PROMPTS

Prompt	Description
Description	Optional description of the exception.
Value Type	The data type defining the value for this exception. Types include Numeric, State, and Time Pattern.
Value	The value to be used during this exception period. The Value Type dictates the information supplied in this field: ■ *Numeric*—a non-negative, real number (e.g., 1) ■ *State*—the name of a State, defined via the States editor dialog box (e.g., On Shift) ■ *Time Pattern*—a time pattern (e.g., Standard Work Day)
Start Date	The calendar date on which this exception should start.
Start Time	The time on the Start Date at which this exception should start.
End Date	The calendar date on which this exception should stop being applied to the resource. If the Value Type is Time Pattern, the values in the time pattern will repeat until this End Date.
End Time	The time on the End Date at which this exception should start being applied.
Exception Type	The type of data to be defined in this time pattern. Types include Capacity and Efficiency.
Application	The list of resources to which this exception should apply.

Calendar States

DESCRIPTION

A Calendar State defines a name that can be used in place of a numeric value when defining time patterns and exceptions. This can be useful if a particular value (e.g., the number of workers in a pool of labor) needs to be entered in multiple places, but may need to be modified. By entering the state name in the time patterns/exceptions, you can modify the value associated with the state, which will apply the new value in all of those places.

States are defined by opening the States editor dialog box from within the time pattern and exceptions editors.

TYPICAL USES

- Labor pool sizes
- Standard efficiency values
- Base arrival rates

PROMPTS

Prompt	Description
Name	The unique name of the calendar state.
Value	The non-negative, real value to be associated with this state.
Color	The color associated with displaying this state.*
Hatch	The hatch pattern associated with displaying this state.*

**Arena does not use the Color and Hatch values; they are utilized by other RSBizWare products.*

4 The Advanced Process Panel

This chapter presents the modules included on the Advanced Process panel, where you'll find additional functionality for modeling your process. Example uses of each module are included in this section.

Flowchart modules

Flowchart modules are the set of objects that are placed in the model window to describe the simulation process.

Delay module

DESCRIPTION

The Delay module delays an entity by a specified amount of time.

When an entity arrives at a Delay module, the time delay expression is evaluated and the entity remains in the module for the resulting time period. The time is then allocated to the entity's value-added, non-value added, transfer, wait, or other time. Associated costs are calculated and allocated as well.

TYPICAL USES

- Processing a check at a bank
- Performing a setup on a machine
- Transferring a document to another department

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Allocation	Type of category to which the entity's incurred delay time and cost will be added.
Delay Time	Determines the value of the delay for the entity.
Units	Time units used for the delay time.

Dropoff module

DESCRIPTION

The Dropoff module removes a specified number of entities from the entity's group and sends them to another module, as specified by a graphical connection. Group user-defined attribute value and internal attributes may be given to the dropped-off entities based on a specified rule.

TYPICAL USES

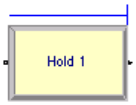
- Loading shelves with product
- Separating a form for use in various departments

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Quantity	Number of entities that will be dropped off from an incoming representative grouped entity.
Starting Rank	Starting rank of the entities to be dropped off, based on the entities in the group.
Member Attributes	Method of determining how to assign the representative entity attribute values (other than costs/times) to the dropped-off original entities.
Attribute Name	Name of representative entity attribute(s) assigned to dropped-off original entities of the group.

Hold module

DESCRIPTION



This module will hold an entity in a queue to wait for a signal, wait for a specified condition to become true (scan), or be held infinitely (to be removed later with the Remove module).

If the entity is holding for a signal, the Signal module is used elsewhere in the model to allow the entity to move on to the next module. If the entity is holding for a given condition to be true, the entity will remain at the module (either in a defined or internal queue) until the condition(s) becomes true. When the entity is in an infinite hold, the Remove module is used elsewhere in the model to allow the entity to continue processing.

TYPICAL USES

- Waiting for a traffic light to turn green
- Holding a part for authorization
- Checking the status of a machine or operator to continue a process

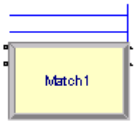
PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Indicates the reasoning for holding the entity within a specified or internal queue. Wait for Signal will hold the entity until a signal of the same value is received. Scan for Condition will hold the entity until the specified condition becomes true. Infinite Hold will hold the entity until it is removed from the queue by a Remove module.
Wait for Value	Signal code for the waiting entity. Applies only when Type is Wait for Signal.
Limit	Maximum number of waiting entities that will be released upon receipt of a signal. Applies only when Type is Wait for Signal.
Condition	Specifies the condition that will be evaluated to hold the entity at the module. If the condition is evaluated to true, the entity leaves the module immediately. If the condition is false, the entity will wait in the associated queue until the condition becomes true. Applies only when Type is Scan for Condition.
Queue Type	Determines the type of queue used to hold the entities. If Queue is selected, the queue name is specified. If Set is selected, the queue set and member in the set are specified. If Internal is selected, an internal queue is used to hold all waiting entities. Attribute and Expression are additional methods for defining the queue to be used.
Queue Name	This field is visible only if Queue Type is Queue, and it defines the symbol name of the queue.
Set Name	This field is visible only if Queue Type is Set, and it defines the queue set that contains the queue being referenced.
Set Index	This field is visible only if Queue Type is Set, and it defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entry for a queue set containing three members is an expression that evaluates to 1, 2, or 3.

Prompt	Description
Attribute	This field is visible only if Queue Type is Attribute. The attribute entered in this field will be evaluated to indicate which queue is to be used.
Expression	This field is visible only if Queue Type is Expression. The expression entered in this field will be evaluated to indicate which queue is to be used.

Match module

DESCRIPTION



The Match module brings together a specified number of entities waiting in different queues. The match may be accomplished when there is at least one entity in each of the desired queues. Additionally, an attribute may be specified such that the entities waiting in the queues must have the same attribute values before the match is initiated.

When an entity arrives at the Match module, it is placed in one of up to five associated queues, based on the entry point to which it is connected. Entities will remain in their respective queues until a match exists.

Once a match exists, one entity from each queue is released to be matched. The matched entities are then synchronized to depart from the module.

TYPICAL USES

- Assembling a part
- Gathering various products for a customer order
- Synchronizing a customer exit with a filled order

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Number to Match	Number of matching entities that must reside in different queues before a match may be completed.
Type	Method for matching the incoming entities. If Type is Any Entities, one entity must reside in each queue for a match to be made. If Type is Based on Attribute, one entity must reside in each queue with the same attribute value.
Attribute Name	Attribute name that is used for identifying an arriving entity's match value. Applies only when Type is Based on Attribute.

Pickup module

DESCRIPTION

The Pickup module removes a number of consecutive entities from a given queue starting at a specified rank in the queue. The entities that are picked up are added to the end of the incoming entity's group.

TYPICAL USES

- Gathering an order from various queue locations
- Gathering completed forms for an office order
- Picking up students at a bus stop for school

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Quantity	Number of entities to pick up.
Queue Name	Name of the queue from which the entities will be picked up, starting at the specified rank.
Starting Rank	Starting rank of the entities to pick up from the queue, Queue Name.

ReadWrite module

DESCRIPTION

The ReadWrite module is used to read data from an input file or the keyboard and assign the data values to a list of variables or attributes (or other expression). This module is also used to write data to an output device, such as the screen or a file.

When reading from or writing to a file, the ReadWrite logic varies according to the *Type of Arena File Name* (the *Access Type* of a file is specified in the File module).

Sequential File or LOTUS Spreadsheet Access Types. When an entity arrives at the ReadWrite module, the specified file is examined to see if it is open (active). If not, the file is automatically opened. The values of the attributes, variables, or expressions (other) listed are read or written according to the format specified.

If you plan to import Lotus (*.wks*) files into Excel, see Lotus 1-2-3 Release 1.x (WKS) Format for guidelines.

Microsoft Excel, Microsoft Access, and ActiveX Data Objects Access Types. When an entity arrives at the ReadWrite module, the specified file is examined to see if an ADO connection to the file is open (active). If the file is currently closed, then an ADO connection is automatically opened to the data source using either the Microsoft Jet OLE DB Provider (if the *Access Type* is *Microsoft Excel* or *Microsoft Access*) or the specified connection string (if the *Access Type* is *ActiveX Data Objects*).

The *Recordset ID* is then examined to see if that recordset in the file is open. A recordset refers to a table of data in the data store that is organized in columns (i.e., fields) and rows (i.e., records).

If the recordset is currently closed, then it is automatically opened using the recordset definition specified in the File module (e.g., the specified named range or table name). The recordset then remains open until the file is closed.

The values of the attributes, variables, or expressions (other) listed in the Assignments repeat group are then read from or written to the specified *Record Number* (i.e., row) in the recordset. The first entry in *Assignments* is read from or written to the first field in the record, the second entry corresponds to the second field, and so on.

Extensible Markup Language Access Type. When an entity arrives at the Read-Write module, the specified file is examined to see if it is open (active). If not, the file is automatically opened into a single ADO recordset.

The values of the attributes, variables, or expressions (other) listed are then read from or written to the specified *Record Number* (i.e., row) in the recordset.

TYPICAL USES

- Reading in airline arrival/departure information
- Prompting an end user for model configuration data (menu)
- Writing order information to a data file, such as order arrival, order fill time, order completion time

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of reading or writing used. Data may be read from a file or from the keyboard. Data may be written to a file or to the screen.
Arena File Name	Name used to identify the file within the File module. Applies only when reading from or writing to a file.

Prompt	Description
Overriding File Format	Format for writing or reading the data. This format will override any format specified in the structure field of the File module. A FORTRAN or C format (enclosed in double quotes) may be used to describe the type and location of each field.
Recordset ID Access Type	Identifies the recordset in Arena File Name that will be accessed. Applies only when the file (specified in the File module) is Microsoft Excel, Microsoft Access, or ActiveX Data Objects.
Record Number Access Type	Record number (i.e., row) in Recordset ID to read from or write to. Applies only when the file (specified in the File module) is Microsoft Excel, Microsoft Access, ActiveX Data Objects, or Extensible Markup Language.
Variable Type	Type of information that will be read or written.
Attribute Name	Defines the symbol name of the attribute to be written or read.
Variable Name	Defines the symbol name of the variable to be read or written.
Row	Specifies the row index for a variable array.
Column	Specifies the column index for a variable array.
Other	Defines the expression for reading or writing other types of information.

Release module

DESCRIPTION

The Release module is used to release units of a resource that an entity previously has seized. This module may be used to release individual resources or may be used to release resources within a set. For each resource to be released, the name and quantity to release are specified.

When the entity enters the Release module, it gives up control of the specified resource(s). Any entities waiting in queues for those resources will gain control of the resources immediately.

TYPICAL USES

- Finishing a customer order (release the operator)
- Completing a tax return (release the accountant)
- Leaving the hospital (release the doctor, nurse, hospital room)

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Type of resource for releasing, either specifying a particular resource, or selecting from a pool of resources (i.e., a resource set). The resource name may also be specified by an expression or attribute value.
Resource Name	Name of the resource that will be released.
Set Name	Name of the resource set from which a member will be released.
Attribute Name	Name of the attribute that specifies the resource name to be released.
Expression	Name of the expression that specifies the name of the resource to be released.
Quantity	Number of resources of a given name or from a given set that will be released. For sets, this value specifies only the number of a selected resource that will be released (based on the resource's capacity), not the number of members to be released within the set.
Release Rule	Method of determining which resource within a set to release. Last Member Seized and First Member Seized will release the last/first member from within the set that was seized. Specific member indicates that a member number or attribute (with a member number value) will be used to specify the member to release.
Set Index	Member index of the resource set that the entity will release.

Remove module

DESCRIPTION

The Remove module removes a single entity from a specified position in a queue and sends it to a designated module.

When an entity arrives at a Remove module, it removes the entity from the specified queue and sends it to the connected module. The rank of the entity signifies the location of the entity within the queue. The entity that caused the removal proceeds to the next module specified and is processed before the removed entity.

TYPICAL USES

- Removing an order from a queue that is due to be completed next
- Calling a patient from a waiting room for an examination
- Retrieving the next order to be processed from a pile of documents

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Queue Name	Name of the queue from which the entity will be removed.
Rank of Entity	Rank of the entity to remove from within the queue.

Seize module

DESCRIPTION

The Seize module allocates units of one or more resources to an entity. The Seize module may be used to seize units of a particular resource, a member of a resource set, or a resource as defined by an alternative method, such as an attribute or expression.

When an entity enters this module, it waits in a queue (if specified) until all specified resources are available simultaneously. Allocation type for resource usage is also specified.

TYPICAL USES

- Beginning a customer order (seize the operator)
- Starting a tax return (seize the accountant)
- Being admitted to hospital (seize the hospital room, nurse, doctor)

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Allocation	Determines to which category the resource usage cost will be allocated for an entity going through the Seize module.
Priority	Priority value of the entity waiting at this module for the resource(s) specified if one or more entities from other modules are waiting for the same resource(s).
Type	Type of resource for seizing, either specifying a particular resource, or selecting from a pool of resources (i.e., a resource set). The name of the resource may also be specified as an attribute value or within an expression.
Resource Name	Name of the resource that will be seized.
Set Name	Name of the resource set from which a member will be seized.
Attribute Name	Name of the attribute that stores the resource name to be seized.

Prompt	Description
Expression	Expression that evaluates to a resource name to be seized.
Quantity	Number of resources of a given name or from a given set that will be seized. For sets, this value specifies only the number of a selected resource that will be seized (based on the resource's capacity), not the number of members to be seized within the set.
Selection Rule	Method of selecting among available resources in a set. **Cyclical** will cycle through available members (e.g., 1-2-3-1-2-3). **Random** will randomly select a member. **Preferred Order** will always select the first available member (e.g., 1, if available; then 2, if available; then 3). **Specific Member** requires an input attribute value to specify which member of the set (previously saved in the Save Attribute field). **Largest Remaining Capacity** and **Smallest Number Busy** are used for resources with multiple capacity.
Save Attribute	Attribute name used to store the index number into the set of the member that is chosen. This attribute can later be referenced with the Specific Member selection rule.
Set Index	Index value into the set that identifies the number into the set of the member requested. If an attribute name is used, the entity must have a value for the attribute before utilizing this option.
Resource State	State of the resource that will be assigned after the resource is seized. The resource state must be defined with the Resource module.
Queue Type	Determines the type of queue used to hold the entities while waiting to seize the resource(s). If Queue is selected, the queue name is specified. If Set is selected, the queue set and member in the set are specified. If Internal is selected, an internal queue is used to hold all waiting entities. Attribute and Expression are additional methods for defining the queue to be used.
Queue Name	This field is visible only if Queue Type is Queue, and it defines the symbol name of the queue.
Set Name	This field is visible only if Queue Type is Set, and it defines the queue set that contains the queue being referenced.
Set Index	This field is visible only if Queue Type is Set, and it defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3.

Prompt	Description
Attribute	This field is visible only if Queue Type is Attribute. The attribute entered in this field will be evaluated to indicate which queue is to be used.
Expression	This field is visible only if Queue Type is Expression. The expression entered in this field will be evaluated to indicate which queue is to be used.

Search module

DESCRIPTION

The Search module searches a queue, a group (batch), or an expression to find the entity rank (for entities in a queue or group) or the value of the global variable *J* that satisfies the specified search condition. When searching a queue or group, the value of the global system variable *J* is set to the rank of the first entity that satisfies Search Condition, or to 0 if Search Condition is not satisfied. When searching an expression, the global system variable *J* is set to the value of the first index value that satisfies the search condition or to zero if no value of *J* in the specified range satisfies the search condition.

When an entity arrives at a Search module, the index *J* is set to the starting index and the search condition is then checked. If the search condition is satisfied, the search ends and the current value of *J* is retained. Otherwise, the value of *J* is increased or decreased and the condition is rechecked. This process repeats until the search condition is satisfied or the ending value is reached. If the condition is not met or there are no entities in the queue or group, *J* is set equal to 0.

TYPICAL USES

- Looking for a particular order number in a queue
- Searching a group for a certain part type
- Determining which process to enter based on availability of resources (search an expression)

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Determination of what will be searched. Search options include entities in a queue, entities within a group (batch) or some expression(s).

Prompt	Description
Queue Name	Name of the queue that will be searched. Applies only when the Type is Search a Queue.
Starting Value	Starting rank in the queue or group or starting value for *J* in an expression.
Ending Value	Ending rank in the queue or group or ending value for *J* in an expression.
Search Condition	Condition containing the index *J* for searching expressions or containing an attribute name(s) for searching queues or batches.

Signal module

DESCRIPTION

The Signal module sends a signal value to each Hold module in the model set to Wait for Signal and releases the maximum specified number of entities.

When an entity arrives at a Signal module, the signal is evaluated and the signal code is sent. At this time, entities at Hold modules that are waiting for the same signal are removed from their queues. The entity sending the signal continues processing until it encounters a delay, enters a queue, or is disposed.

TYPICAL USES

- Analyzing traffic patterns at an intersection (signal when the light turns green)
- Signaling an operator to complete an order that was waiting for a component part

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Signal Value	Value of the signal to be sent to entities in Hold modules.
Limit	Maximum number of entities that are to be released from any Hold modules when the signal is received.

Store module

DESCRIPTION

The Store module adds an entity to storage. The Unstore module may then be used to remove the entity from the storage.

When an entity arrives at the Store module, the storage specified is incremented, and the entity immediately moves to the next module in the model.

Storages are useful for displaying entity animation while an entity undergoes processing in other modules. Additionally, statistics may be kept on the number of entities in storage.

TYPICAL USES

- Animating a part through a number of delay operations (load, setup, process, unload)
- Tracking the number of customers within a grocery store (place in storage upon entry)

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of specifying the storage name as a **Storage**, **Set**, **Attribute**, or **Expression**.
Storage Name	Name of the storage to which the entity will be added. Applies only when the Type is Storage.
Set Name	Name of the storage set from which the storage is to be selected. Applies only when the Type is Set.
Set Index	Index into the defined storage set that contains the desired storage name. Applies only when the Type is Set.
Attribute	Name of the attribute whose value contains the storage. Applies only when the Type is Attribute.
Expression	Expression that is evaluated to the storage into which the entity is placed. Applies only when the Type is Expression.

Unstore module

DESCRIPTION

The Unstore module removes an entity from storage. When an entity arrives at the Unstore module, the storage specified is decreased and the entity immediately moves to the next module in the model.

TYPICAL USES

- Removing the entity from an animation location when processing is complete
- Tracking the number of customers within a grocery store (unstore upon exit)

PROMPTS

Prompt	Description
Name	Unique module identifier displayed on the module shape.
Type	Method of specifying the storage name as a **Storage**, **Set**, **Attribute**, or **Expression**. Default will remove an entity from the last storage that it entered.
Storage Name	Name of the storage to which the entity will be added. Applies only when the Type is Storage.
Set Name	Name of the storage set from which the storage is to be selected. Applies only when the Type is Set.
Set Index	Index into the defined storage set that contains the desired storage name. Applies only when the Type is Set.
Attribute	Name of the attribute whose value contains the storage. Applies only when the Type is Attribute.
Expression	Expression that is evaluated to the storage into which the entity is placed. Applies only when the Type is Expression.

Adjust Variable module

DESCRIPTION

The Adjust Variable module adjusts a variable to a target value at a specified rate. This module might be used with the *Rotate By Expression* option in Global and Resource pictures to present smooth animated rotations of pictures. It might also be used to approximate/animate a continuous increase or decrease to a variable value over time.

When an entity enters an Adjust Variable module, the *Variable Name* is adjusted to the *To Value* at the specified *Rate*. The entity is held in the module until the adjustment is completed.

The *Update Interval* specifies the time interval between actual assignment updates of the variable value over the time period of the change. A smaller update interval produces smoother animation in plots or rotating pictures that reference the variable, as well as more accurate time-persistent statistics collected on the variable. A larger update interval produces faster run speeds.

TYPICAL USES

- Rotate a resource picture to 180 degrees at a rate of 5 degrees per second

PROMPTS

Prompt	Description
Name	Unique module identifier. This name is displayed on the module shape.
Variable Type	Specifies the array characteristics of the variable being adjusted.
Row	Specifies the row index if the variable is an array.
Column	Specifies the column index if the variable is a 2-D array.
Variable Name	Specifies the name of the variable to adjust.
To Value	Specifies the target value that the variable will be adjusted to.
Rate	Specifies the rate at which the adjustment will be made.
Units	The time units of the rate.
Update Interval	Specifies the interval between updates of the variable's value as the variable is adjusted at the specified rate.
Units	The time units of the update interval.
Allocation	Type of category to which the entity's incurred delay time and cost in the module will be added.

Data modules

Data modules are the set of objects in the spreadsheet view of the model that define the characteristics of various process elements, such as resources and queues.

Advanced Set module

DESCRIPTION

The Advanced Set module specifies queue sets, storage sets, and other sets and their respective members. A set defines a group of similar elements that may be referenced via a common name and a set index. The elements that make up the set are referred to as the members of the set.

Queue sets may be specified within a Seize or material-handling-type module. Storage sets may be used in the Store and Unstore module. Other sets may include sets of attributes or other miscellaneous elements.

TYPICAL USES

- Various checkout queues in a grocery store (queues)
- Animation locations for performing operations (storages)

PROMPTS

Prompt	Description
Name	The name of the advanced set whose members are being defined. This name must be unique.
Set Type	Type of set that is being defined, which may include **Queue**, **Storage**, or **Other** types of sets.
Queue Name	Name of the queues that are included within a queue set.
Storage Name	Name of the storages that are included within a storage set.
Other	Name of the members that are included within an "other" type set.

Expression module

DESCRIPTION

The Expression module defines expressions and their associated values. Expressions are referenced in the model by using their name. Expressions can optionally be specified as one- or two-dimensional arrays.

An expression value may be formed using combinations of integers, real numbers, symbol names, statistical distributions (e.g., NORM(10,2)), arithmetic operators (e.g., +, *), parentheses, logical operators, (e.g., .GT. or >), attributes, and variables.

An expression value may be referenced in another expression, but recursive calls to the same expression are not allowed.

TYPICAL USES

- Distribution for an entity arrival
- Entity picture values
- Complex expression for order entry times

PROMPTS

Prompt	Description
Name	The name of the expression whose characteristics are being defined. This name must be unique.
Row	Maximum number of rows in the defined expression.
Column	Maximum number of columns in the defined expression. Visible only when the number of rows is specified.
Expression Value	Expression value associated with the expression name.

Failure module

DESCRIPTION

The Failure module is designed for use with resources. When a failure occurs, the entire resource (regardless of its capacity) is failed. Failures are designed to be used with single-capacity resources or with multiple-capacity resources whose individual resource units all fail at the same time.

TYPICAL USES

- Breakdown information for a machine
- Cash register tape refill every “x” customers
- Random computer shutdowns or “reboots”

PROMPTS

Prompt	Description
Name	The name of the failure associated with one or more resources.
Type	Determines if the failure is **time-based** or **count-based**.
Count	Defines the number of resource releases for count-based failures. Valid when the Type is Count.
Up Time	Defines the time between failures for time-based failures. Valid when the Type is Time.
Up Time Units	Time units for the time between failures (Up Time) for time-based failures.
Down Time	Defines the duration of the failure.
Down Time Units	Time units for the duration of the failure (Down Time).

Prompt	Description
Uptime in this State only	Defines the state that should be considered for the time between failures (only for time-based failures). If state is not specified, then all states are considered (i.e., the time between failures does not depend on the time spent in a specific state, but rather on the total simulation time). For example, you might want to define a failure to be based only on the state Busy, and therefore, the time between downtimes would be based on the amount of time that a resource is busy, not simulated clock time.

File module

DESCRIPTION

The File module must be included whenever external files are accessed using the Read-Write module. This module identifies the system file name and defines the access method, formatting, and operational characteristics of the files.

TYPICAL USES

- File containing predefined airline flight data
- File specifying customer order times and relevant information
- File to write user model configuration data from menu input

PROMPTS

Prompt	Description
Name	The name of the file whose characteristics are being defined. This name must be unique.
Access Type	The file type.
Operating System File Name	Name of the actual file that is being read from or to which it is being written.
Connecting String	Connection string used to open ADO connection to the data source.
Structure	File structure, which can be unformatted, free format, or a specific C or FORTRAN format.
End of File Action	Type of action to occur if an end of file condition is reached.
Initialize Option	Action to be taken on file at beginning of each simulation replication.
Comment	Character indicating comment record.

Prompt	Description
Recordset Name	Name used to identify the recordset in ReadWrite module. This name must be unique within the file.
Command Text	Text of the command that will be used to open the recordset (e.g., SQL statement, procedure name, table name.)
Command Type	Type of command entered in the CommandText.
Named Range	The named range in the Excel workbook to which the recordset refers.
Table Name	The name of the table in the Access database to which the recordset refers.

StateSet module

DESCRIPTION

The StateSet module is used to define states for a resource or number of resources. The states may be associated with an autostate or may be new states for the resource. The Resource module in the Basic Process panel references the stateset, if any, that a given resource will use.

TYPICAL USES

- States for an operator, such as busy with customer, busy refilling shelves, busy on phone
- States for a machine, such as random failure, scheduled failure, busy with setup, busy with processing
- States for a doctor, such as scheduled appointments, hospital rounds, paperwork, telephone calls

PROMPTS

Prompt	Description
StateSet Name	Name of set of states that the resource may be assigned during the simulation run.
State Name	Name of the user-defined state.
AutoState or Failure	Used to associate the state name with an autostate (such as **Idle**, **Busy**, **Inactive**, or **Failed**) or a particular Failure Name. If not used, the state must be assigned by using either the Assign module or the Resource State within the Seize or Process modules.

Statistic module

DESCRIPTION

The Statistic module is used to define additional statistics that are to be collected during the simulation and also to specify output data files. While summary statistics (e.g., Average and Maximum) are automatically generated for each statistic, if you specify an output file, then each individual observation (e.g., each tally recorded) is written out to an output file. The types of statistics that may be defined in the Statistic module are time-persistent, tallies (observational data), count-based, outputs, and frequency-based.

TYPICAL USES

- Gathering observational data to compare two simulation configurations (save data to file for a tally)
- Collecting additional statistics information such as number in queue plus number resources busy
- Generating statistics on various states of an operator or machine
- Tracking the number of customers in a grocery store using NSTO function and Store/Unstore modules

PROMPTS

Prompt	Description
Name	The name of the schedule being defined. This name must be unique.
Type	Type of statistic being defined. Type may be **Time-Persistent** (Dstat), observational **Tally-**, **Counter-**, **Output-** or **Frequency-**based.
Tally Name	Defines the symbol name for the tally. This name appears in the report as the tally identifier. Applies only when Type is Tally.
Counter Name	Defines the symbol name for the counter. This name appears in the report as the counter identifier. Applies only when Type is Counter.
Expression	Name of the time-persistent statistic, value of the output statistic or expression for the frequency. For frequencies, if the Frequency Type is State, this field is a resource name indicating that statistics are to be collected on the state of a resource. If Frequency Type is Value, this field simply specifies the expression on which statistics are to be collected. Applies only when Type is Time-Persistent, Output, or Frequency.
Frequency Type	Indicates whether statistics are being collected on the state of a resource (State) or on an expression (Value). Applies only when Type is Frequency.

Prompt	Description
Report Label	Defines the symbol name for the expression. This name appears as the report label associated with the expression in the reports. If a statistic has already been defined through a different module, the Report Label field will automatically appear. Applies only when Type is Time-Persistent, Output, or Frequency.
Limit	May be used to define the terminating condition for a simulation replication. When the counter reaches or exceeds the specified limit, the simulation replication ends and the report is generated. Applies only when Type is Counter.
Initialization Option	Indicates if the counter will be initialized (reset to zero) between simulation replications. If the field is specified as **No** and multiple replications are performed, then the value of the counter at the end of a replication will be retained as the initial value at the beginning of the next replication. When specified as **Yes**, the counter value is set to 0 at the beginning of each replication. When set to **Replicate**, the counter value will be cleared when the other statistics (e.g., tallies, time-persistent) are cleared as specified in Setup.
Tally Output File	Specifies the system-specific file name to which the raw statistical data are written. Applies only when Type is Tally.
Counter Output File	Specifies the system-specific file name to which the raw statistical data are written. Applies only when Type is Counter.
Output File	Specifies the system-specific file name to which the raw statistical data are written. Applies only when Type is Frequency, Output, or Time-Persistent.
Constant or Range	Specifies whether statistics are being collected based on the Expression Value being a Constant value (or state of a resource) or a Range of values.
Value	Specifies the value when collecting statistics for a Constant value. Therefore, statistics will be collected against this category whenever the Expression Value equals this Value (or whenever the resource is in this state). If collecting statistics for a Range value, this field specifies the lowest value in the range. Therefore, statistics will be collected against this category whenever the Expression Value is greater than Value and less than or equal to High Value.
High Value	Specifies the highest value in the range, if collecting statistics for a Range value. Statistics will be collected against this category whenever the Expression Value is greater than Value and less than or equal to High Value.

Prompt	Description
Category Name	Defines the symbol name for this category. This name appears in the summary report as the category identifier. If left blank, the identifier will simply be the Value or Range.
Category Option	This field indicates whether or not a particular category is to be excluded from the Restricted Column in the summary report.

Storage module

DESCRIPTION

The Storage module defines the name of a storage. Storages are automatically created by any module that references the storage so that this module is seldom needed. The only time this module is needed is when a storage is defined as a member of a storage set or specified using an attribute or expression.

TYPICAL USES

- Defining an animate storage for a set of storages

PROMPTS

Prompt	Description
Name	The name of the storage set being defined. This name must be unique.

5 The Advanced Transfer Panel

This chapter presents the modules from the Advanced Transfer panel, which includes the modules needed for modeling the movement of entities from one location to another. Example uses of each module are included in this section.

General flowchart modules

Flowchart modules are the set of objects that are placed in the model window to describe the simulation process.

Enter module

DESCRIPTION

The Enter module defines a station (or a set of stations) corresponding to a physical or logical location where processing occurs. When an entity arrives at an Enter module, an unloading delay may occur and any transfer device used to transfer the entity to the Enter module's station may be released.

The station (or each station within the defined set) has a matching Activity Area that is used to report all times and costs accrued by the entities in this station. This Activity Area's name is the same as the station. If a parent Activity Area is defined, then it also accrues any times and costs by the entities in this station.

TYPICAL USES

- The start of a part's production in a series of parallel processes where the part's forklift needs to be released
- The start of a document's processing after the document has been created where the mail clerk resource needs to be released

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Station Type	Type of station, either a single **Station** or station **Set**.
Station Name	Name of the individual station. A given station can only exist once within a model.
Parent Activity Area	Name of the Activity Area's parent.
Associated Intersection	Name of the intersection associated with this station in a guided transporter network.

Prompt	Description
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this station and its corresponding activity area.
Set Name	Name of the station set. A given station set can only exist once within a model.
Save Attribute	Specifies the attribute to be used to store the index into the station set for an entity entering this module.
Set Members	This repeat group permits you to define the individual stations that are to be members of the specified station set. A station set must have at least one member station. Active when Station Type is Set.
Station Name	This field indicates the name of a station that is to be a member of this station set. A given station can only exist within a model once. Therefore, an individual station can only be the member of one station set, and that individual station may not be the name of a station in another module.
Parent Activity Area	Name of the Activity Area's parent for the station set member.
Associated Intersection	Name of the intersection associated with this station set in a guided transporter network.
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this station set member and its corresponding activity area.
Allocation	Type of category to which the entity's incurred delay time and cost will be added.
Delay	This field defines the delay that will be experienced by entities immediately upon arrival at the station.
Units	Time units used for the delay time.
Transfer In	If a resource, transporter, or conveyor was used to transfer the entity to this station, this can be used to release, free, or exit the device. If **Release Resource** is selected, the specified resource is released. If **Free Transporter** is selected, the specified transporter is freed. If **Exit Conveyor** is selected, the specified conveyor is exited.
Transporter Name	Name of the transporter to be freed upon arrival to the station. Active when Transfer Name is Free Transporter.
Unit Number	Unit number of the transporter if the transporter is multi-capacity.
Conveyor Name	Name of the conveyor to exit upon arrival to the station.

Prompt	Description
Resource Type	Type of allocation, either single **Resource** or resource **Set**.
Resource Name	Name of the resource to release. Active when Transfer Name is Release Resource.
Set Name	Name of the resource set from which the resource is to be released.
Release Rule	Determines which member of the set is to be released, either the **Last Member Seized**, **First Member Seized**, or **Specific Member**.
Set Index	Index into the set that determines which member of the set is to be released.
Attribute Name	Name of the attribute that determines the instance number of the resource to release.
Expression	Expression value that determines the instance number of the resource to release.

Leave module

DESCRIPTION

The Leave module is used to transfer an entity to a station or module. An entity may be transferred in two ways. It can be transferred to a module that defines a station by referencing the station and routing, conveying, or transporting to that station, or a graphical connection can be used to transfer an entity to another module. When an entity arrives at a Leave module, it may wait to obtain a transfer device (resource, transporter, or conveyor). When the transfer device has been obtained, the entity may experience a loading delay. Finally, the entity is transferred from this module to a destination module or station.

TYPICAL USES

- The end of a part's production in a series of parallel processes where the part needs a forklift to be transferred to shipping

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Allocation	Type of category to which the entity's incurred delay time and cost will be added.
Delay	Specifies a load time incurred after getting a transfer device.

Prompt	Description
Units	Time units used for the delay time.
Transfer Out	Determines whether a resource (**Seize Resource**), transporter (**Request Transporter**), or conveyor (**Access Conveyor**) is required prior to transferring the entity out of this module.
Priority	Indicates the priority of the module when either seizing a resource or requesting a transporter when there are entities waiting for that resource/transporter from other modules. This field is not visible when the Transfer Type is None or Access Conveyor.
Transporter Name	Name of the transporter to request.
Queue Type	Type of queue, either a single **Queue**, queue **Set**, **Internal** queue, **Attribute**, or **Expression**.
Queue Name	Name of the individual queue.
Queue Set Name	Name of the queue set.
Set Index	Defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set.
Queue Attribute Name	The attribute name that will be evaluated to indicate which queue is to be used.
Queue Expression	The expression that will be evaluated to indicate which queue is to be used.
Selection Rule	Method of selecting among available transporters in a set. Cyclical will cycle through available members. **Random** will randomly select a member. **Preferred Order** will always select the first available member. **Specific Member** requires an input attribute value to specify which member of the set (previously saved in the Save Attribute field). **Largest Distance** selects the transporter farthest away, and **Smallest Distance** selects the closest transporter.
Save Attribute	Attribute name used to store the index number into the set of the member that is chosen. This attribute can later be referenced with the **Specific Member** selection rule. Active when Transfer Out is Request Transporter.
Index Set	Attribute name whose value identifies the index number into the set of the member requested. The entity must have a value for the attribute before utilizing this option.
Resource Type	Type of resource for seizing, either specifying a particular **Resource**, selecting from a pool of resources (i.e., a resource **Set**), **Attribute**, or **Expression**.

Prompt	Description
Resource Name	Name of the resource to seize.
Conveyor Name	Name of the conveyor to access.
# of Cells	Number of contiguous cells the entity requires.
Connect Type	Determines if the entity is to **Route**, **Convey**, or **Transport** to another station or **Connect** to another module.
Move Time	Time to route from this module to the destination station.
Units	Time units used for the move time.
Station Type	The entity's destination station type either an individual **Station**, a station based on an **Attribute** or **Expression** value, or **By Sequence.**
Station Name	Name of the individual destination station.
Attribute Name	The attribute name that will be evaluated to indicate the station.
Expression	The expression that will be evaluated to indicate the station.

PickStation module

DESCRIPTION

The PickStation module allows an entity to select a particular station from the multiple stations specified. This module picks among the group of stations based on the selection logic defined with the module. The entity may then route, transport, convey, or connect to the station specified. If the method chosen is connect, the selected station is assigned to an entity attribute. The station selection process is based on the minimum or maximum value of a variety of system variables and expressions.

TYPICAL USES

- A part sent to a processing station based on machine's availability at each station
- A loan application sent to a set of loan officers based on the number sent to each officer
- A customer selecting among cashier lines based on the least number waiting in each line

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.

Prompt	Description
Test Condition	Test condition to use for the station selection process, either **Minimum** or **Maximum**.
Number En Route to Station	The number of entities transferring to the station is considered in the station selection process.
Number in Queue	The number of entities in the queue at the station is considered in the station selection process.
Number of Resources Busy	The number of busy resources at the station is considered in the station selection process.
Expression	Determines if an additional user-defined expression is considered in the station selection process.
Transfer Type	Determines how an entity will be transferred out of this module to its next destination station—either **Route**, **Convey**, **Transport**, or **Connect**.
Save Attribute	Defines the name of the attribute that will store the station name that is selected, visible when the transfer method is Connect.
Route Time	Move time of the entity from its current station to the station determined through this module.
Units	Time units for route-time parameters.

PickStation module — Stations dialog

PROMPTS

Prompt	Description
Station Name	Station name for the selection process. All potential station names are required even though the Number En Route to station may not be a consideration in the PickStation evaluation.
Queue Name	Visible only if Number in Queue is checked, it defines the name of the queue corresponding to the station defined.
Resource Name	Visible only if Number of Resources Busy is checked, it defines the resource name corresponding to the station defined.
Expression	Visible only if Expression is checked, it defines an additional expression that will be considered in the station selection process.

Route module

DESCRIPTION

The Route module transfers an entity to a specified station or the next station in the station visitation sequence defined for the entity. A delay time to transfer to the next station may be defined.

When an entity enters the Route module, its Station attribute (Entity.Station) is set to the destination station. The entity is then sent to the destination station, using the route time specified.

If the station destination is entered as By Sequence, the next station is determined by the entity's sequence and step within the set (defined by special-purpose attributes Entity.Sequence and Entity.Jobstep, respectively).

TYPICAL USES

- Send a part to its next processing station based on its routing slip
- Send an account balance call to an account agent
- Send restaurant customers to a specific table

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Route Time	Travel time from the entity's current location to the destination station.
Units	Time units for route-time parameters.
Destination Type	Method for determining the entity destination location. Selection of **By Sequence** requires that the entity has been assigned a sequence name and that the sequence itself has been defined.
Station Name	Name of the individual destination station.
Attribute Name	Name of the attribute that stores the station name to which entities will route.
Expression	Expression that is evaluated to the station name where entities will route.

Station module

DESCRIPTION

The Station module defines a station (or a set of stations) corresponding to a physical or logical location where processing occurs. If the Station module defines a station set, it is effectively defining multiple processing locations.

The station (or each station within the defined set) has a matching Activity Area that is used to report all times and costs accrued by the entities in this station. This Activity Area's name is the same as the station. If a parent Activity Area is defined, then it also accrues any times and costs by the entities in this station.

TYPICAL USES

- Defining a lathe area
- Defining a set of toll booths
- Defining a food preparation area

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Station Type	Type of station being defined, either as an individual **Station** or a station **Set**.
Station Name	Name of the individual station.
Set Name	Name of the station set.
Parent Activity Area	Name of the Activity Area's parent.
Associated Intersection	Name of the intersection associated with this station in a guided transporter network.
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this station and its corresponding activity area.
Save Attribute	Attribute name used to store the index number into the station set of the member that is selected.
Station Set Members	Names of the stations that are members of this station set.
Station Name	A given station can only exist once within a model. Therefore, an individual station can only be the member of one station set, and that individual station may not be the name of a station in another module.
Parent Activity Area	Name of the Activity Area's parent for the station set member.

Prompt	Description
Associated Intersection	Name of the intersection associated with this station set in a guided transporter network.
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this station set member and its corresponding activity area.

Conveyor flowchart modules

Access module

DESCRIPTION

The Access module allocates one or more cells of a conveyor to an entity for movement from one station to another. Once the entity has control of the cells on the conveyor, it may then be conveyed to the next station.

When an entity arrives at an Access module, it will wait until the appropriate number of contiguous cells on the conveyor are empty and aligned with the entity's station location.

TYPICAL USES

- Parts accessing a conveyor to be sent to a paint booth
- Glass accessing a conveyor to be transferred to a cutting station

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor that the entity desires.
# of Cells	Number of contiguous conveyor cells the entity requires for movement on the conveyor.
Queue Type	Determines the type of queue used to hold the entities, either an individual **Queue**, a queue **Set**, and **Internal** queue or an **Attribute** or **Expression** that evaluate to the queue name.
Queue Name	Name of the queue that will hold the entity until it accesses the conveyor.
Set Name	Name of the set of queues.

Prompt	Description
Set Index	Defines the index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3.
Attribute Name	Defines the name of the attribute that stores the queue name to which entities will reside.
Expression	Defines the name of the expression that stores the queue name to which entities will reside.

Convey module

DESCRIPTION

The Convey module moves an entity on a conveyor from its current station location to a specified destination station. The time delay to convey the entity from one station to the next is based on the velocity of the conveyor (specified in the Conveyor module) and the distance between the stations (specified in the Segment module).

When an entity enters a Convey module, its Station attribute (*Entity.Station*), is set to the destination station. The entity is then conveyed to the destination station. If the destination type is specified as By Sequence, the next station is determined by the entity's Sequence and Jobstep within the sequence (special-purpose attributes *Entity.Sequence* and *Entity.Jobstep*, respectively).

TYPICAL USES

- Convey bags from a location to the baggage claim area
- Convey parts from a load station to processing station

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor on which the entity will convey. If left blank, the previously accessed conveyor is assumed.
Destination Type	Method for specifying the entity destination (**By Sequence**, **Station**, **Attribute**, or **Expression**). Selection of By Sequence requires that the entity has been assigned a sequence name and that the sequence itself has been defined.

Prompt	Description
Station Name	Name of the individual station.
Attribute Name	Attribute that stores the station name to which entities will route.
Expression	Expression that is evaluated to the station name where entities will route.

Exit module

DESCRIPTION

The Exit module releases the entity's cells on the specified conveyor. If another entity is waiting in queue for the conveyor at the same station when the cells are released, it will then access the conveyor.

TYPICAL USES

- Cases exit a conveyor for packing
- Bad parts are removed from the conveyor and disposed
- Passengers remove luggage from the baggage claim conveyor

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor on which the entity will exit. If left blank, the previously accessed conveyor is assumed.
# of Cells	Number of contiguous conveyor cells the entity will relinquish.

Start module

DESCRIPTION

The Start module changes the status of a conveyor from inactive to active. The conveyor may have been deactivated from either the Stop module or by initially being set to inactive at the start of the simulation. The velocity of the conveyor may be changed permanently when the conveyor is started.

TYPICAL USES

- Start a bottling conveyor after scheduled maintenance
- Start a baggage claim conveyor when bags have arrived

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor to start.
Velocity	Speed of the conveyor once it begins to operate. This value will change the speed of the conveyor permanently, until it is changed in another module.
Units	Velocity time units.

Stop module

DESCRIPTION

The Stop module sets the operational status of a conveyor to inactive. The conveyor may have been activated from either the Start module or by initially being set to active at the start of the simulation. When the entity enters the Stop module, the conveyor will stop immediately, regardless of the type of conveyor or the number of entities currently on the conveyor.

TYPICAL USES

- Stop a baggage conveyor after a pre-determined amount of time
- Stop a conveyor for scheduled maintenance

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Conveyor Name	Name of the conveyor to stop.

Transporter flowchart modules

Activate module

DESCRIPTION

The Activate module increases the capacity of a previously halted transporter. The transporter unit that is activated will reside at the station location at which it was halted until it is moved or requested by an entity. If there is an entity(s) awaiting a transporter at the time the unit is activated, the entity will gain control of the transporter immediately.

TYPICAL USES

- Activate a forklift that was down for scheduled maintenance
- Activate gurneys in an emergency room after cleaning and restocking
- Start additional wait staff at a restaurant

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to activate.
Unit Number	Determines which of the transporter units in the transporter set to activate.

Allocate module

DESCRIPTION

The Allocate module assigns a transporter to an entity without moving it to the entity's station location. The entity then has control of the transporter to either move it to a particular location or to halt it for a breakdown or failure. A particular transporter unit may be allocated to the entity or a selection rule may be used to determine which of the transporters will be assigned to the entity.

TYPICAL USES

- A mechanic allocates a forklift for scheduled maintenance
- Allocate a taxi to pick up waiting passengers

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to allocate.
Selection Rule	Rule for determining which of the transporters to allocate to the entity. The selection rule has the following options: **Cyclical**, **Random**, **Preferred Order**, **Specific Member**, **Largest Distance**, and **Smallest Distance**.
Save Attribute	The attribute name that will store the unit number of the selected transporter.
Unit Number	Determines the specific transporter in the transporter set to allocate to the entity. It is only visible if selection rule is Specific Member.
Priority	Priority value of the entity waiting at this module for the transporter specified if one or more entities are waiting for the same transporter anywhere in the model.
Entity Location	This field is used to determine the transporter closest to or farthest from the requesting entity when using the Largest Distance or Smallest Distance transporter selection rules. For free-path transporters, the entity location must be specified as a station. For guided transporters, the entity location may be specified as a station, intersection, or network link.
Queue Type	Type of queue used to hold the entities while waiting to allocate the transporter, either an individual **Queue**, a queue **Set**, an **Internal** queue, or an **Attribute** or **Expression** that evaluate to the queue name.
Queue Name	Name of the individual queue.
Queue Set Name	Name of the queue set that contains the queue being referenced.
Set Index	The index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3.
Attribute Name	The attribute name that will be evaluated to the queue name.
Expression	The expression that will be evaluated to the queue name.

Free module

DESCRIPTION

The Free module releases the entity's most recently allocated transporter unit. If another entity is waiting in a queue to request or allocate the transporter, the transporter will be given to that entity. If there are no waiting entities at the time the transporter unit is freed, the transporter will wait idle at the freeing entity's station location, unless otherwise specified in the Transporter module.

TYPICAL USES

- A part awaiting a shipping truck frees its forklift
- An airport transfer cart completes its trip

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to free. A blank value assumes the most recently allocated or requested transporter.
Unit Number	Determines which of the transporter units in the transporter set to free.

Halt module

DESCRIPTION

The Halt module changes the status of a transporter unit to inactive. If the transporter is busy at the time when an entity enters the Halt module, the status is considered busy and inactive until the entity that controls the transporter frees the unit. If the transporter is idle at the time when an entity halts the transporter, it is set to inactive immediately. Once a transporter unit has been halted, no entities will get control of the transporter until it is activated.

TYPICAL USES

- Stop a forklift for scheduled maintenance
- Disable a broken gurney in an emergency room

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to halt.
Unit Number	Determines which of the transporter units in the transporter set to halt.

Move module

DESCRIPTION

The Move module advances a transporter from one location to another without moving the controlling entity to the destination station. The controlling entity remains at its current module location until the transporter arrives at its destination. At that time, the entity will be able to move to another module or task in the model.

If the transporter being moved is a free-path transporter, then the time delay to move the transporter from one station to the next is based on the velocity of the transporter and the distance between the stations (specified in the Distance module). If the transporter being moved is a guided transporter, then the time delay to move the transporter depends not only on the velocity and distance to the destination, but also the transporter's acceleration, deceleration, and any traffic it encounters on the way.

TYPICAL USES

- Move a broken forklift to a service station
- Move a worker to a break room
- Move a waiter to the kitchen

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to move.
Unit Number	Determines which of the transporter units in the transporter set to move.
Destination Type	Determines the method for specifying the transporter destination. The **Intersection** and **Network Link** options apply only to guided transporters.
Station Name	Name of the individual destination station.

Prompt	Description
Attribute Name	Name of the attribute that stores the destination station name to which entities will route.
Expression	Expression that is evaluated to the destination station name where entities will route.
Intersection Name	Defines the name of the intersection to which the guided transporter will move.
Network Link Name	Defines the name of the network link to which the guided transporter will move.
Zone	The specific zone number of the Network Link Name.
Velocity	Specifies the temporary velocity at which the transporter is moved to the destination station.
Units	Velocity time units.

Request module

DESCRIPTION

The Request module assigns a transporter unit to an entity and moves the unit to the entity's station location. A specific transporter unit may be specified or the selection may occur based on a rule.

When the entity arrives at the Request module, it is allocated a transporter when one is available. The entity remains at the Request module until the transporter unit has reached the entity's station. The entity then moves out of the Request module.

TYPICAL USES

- A sanded part requests a cart to take it to the paint shop
- Customers in a restaurant are ready to order and thus request a waiter

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to allocate.
Selection Rule	Rule for determining which of the transporters to allocate to the entity, either **Cyclical, Random**, **Preferred Order**, **Specific Member**, **Largest Distance**, and **Smallest Distance**.

Prompt	Description
Save Attribute	Denotes the attribute name that will store the unit number of the selected transporter.
Unit Number	Determines the specific transporter unit in the transporter set to request.
Priority	Priority value of the entity waiting at this module for the transporter specified if one or more entities are waiting for the same transporter anywhere in the model.
Entity Location	The location to which the transporter will move after being allocated. For free-path transporters, the entity location must be specified as a station. For guided transporters, the entity location may be specified as a station, intersection, or network link.
Velocity	Speed at which the transporter is moved to the entity location in length units per unit time. The unit time is specified in the Units field.
Units	Velocity time units.
Queue Type	Type of queue used to hold the entities while waiting to access the transporter, either an individual **Queue**, a queue **Set**, an **Internal** queue, or an **Attribute** or **Expression** that evaluate to the queue's name.
Queue Name	Name of the individual queue.
Set Name	Name of the queue set.
Set Index	The index into the queue set. Note that this is the index into the set and not the name of the queue in the set. For example, the only valid entries for a queue set containing three members is an expression that evaluates to 1, 2, or 3.
Attribute Name	The attribute name that will be evaluated to the queue name.
Expression	Expression evaluated to the queue name.

Transport module

DESCRIPTION

The Transport module transfers the controlling entity to a destination station. After the time delay required for the transport, the entity reappears in the model at the Station module. Free-path transporters are always moved to the same station destination as the entity. Guided transporters may be moved to a different network location than the entity's station destination using the Guided Tran Destination Type.

If a free-path transporter is being used, then the time delay to move the entity and transporter from one station to the next is based on the velocity of the transporter and the distance between the stations (specified in the Distance module). If a guided transporter is being used, then the time delay to transport the entity depends not only on the velocity and distance to the guided transporter's destination, but also the transporter's acceleration, deceleration, and any traffic it encounters on the way.

When an entity enters the Transport module, its Station attribute (Entity.Station) is set to the entity's destination station. The entity is then transported to the destination station.

If the station destination is entered as By Sequence, the next station is determined by the entity's Sequence and Jobstep within the set (special-purpose attributes Entity.Sequence and Entity.Jobstep, respectively).

TYPICAL USES

- A forklift transports a pallet of parts to the next processing station
- A mail cart transports packages from the mailroom to the order-processing station

PROMPTS

Prompt	Description
Name	Unique name of the module that will be displayed in the flowchart.
Transporter Name	Name of the transporter to move. If blank, the most recently requested or allocated transporter is assumed.
Unit Number	Determines which of the transporter units in the transporter set to transport.
Entity Destination Type	Determines the method for specifying the entity destination station logic. Selection of **By Sequence** requires that the entity has been assigned a sequence name and that the sequence itself has been defined.
Station Name	Name of the individual station.
Attribute Name	Name of the attribute that stores the station name to which entities will route.
Expression	Expression that is evaluated to the station where entities will route.
Velocity	Specifies the temporary velocity at which the entity and transporter are moved to the destination station.
Units	Velocity time units.
Guided Tran Destination Type	Allows the specification of a guided transporter destination that is different from the entity's destination. This field is ignored if Transporter Name is a free-path transporter.

Prompt	Description
Station Name	Defines the name of the station with an associated intersection to which the guided transporter will move.
Attribute Name	Defines the name of the attribute that stores the station name with an associated intersection to which the guided transporter will move.
Expression	Defines an expression that evaluates to the network location (intersection, station, network link) to which the guided transporter will move.
Intersection Name	Defines the name of the intersection to which the guided transporter will move.
Network Link Name	Defines the name of the network link to which the guided transporter will move.
Zone	The specific zone number on Network Link Name.

Data modules

Data modules are the set of objects in the spreadsheet view of the model that define the characteristics of various process elements, such as resources and queues.

Sequence module

DESCRIPTION

The Sequence module is used to define a sequence for entity flow through the model. A sequence consists of an ordered list of stations that an entity will visit. For each station in the visitation sequence, attributes and variables may be assigned values.

Each station in the visitation sequence is referred to as a step (or jobstep) in the sequence.

Three special-purpose attributes are provided for all entities. The Sequence attribute (Entity.Sequence) defines the sequence that an entity is to follow; a value of 0 indicates that the entity is not following any sequence. In order for an entity to follow a sequence, its Sequence attribute must be assigned a value (e.g., in the Assign module).

The Jobstep attribute (Entity.Jobstep) stores the entity's current step number in the sequence. This value is updated automatically each time an entity is transferred. You typically do not need to assign explicitly a value to Jobstep in the model.

The PlannedStation attribute (Entity.PlannedStation) stores the number of the station associated with the next jobstep in the sequence. This attribute is not user-assignable. It is

automatically updated whenever Entity.Sequence or Entity.JobStep changes, or whenever the entity enters a station.

Jobstep names must be globally unique.

TYPICAL USES

- Define a routing path for part processing
- Define a sequence of steps patients must take upon arrival at an emergency room

PROMPTS

Prompt	Description
Name	Name of the sequence.
Steps	Repeat group that defines the ordered list of stations an entity visits for the sequence named, as well as attribute and variable assignments to be made at each of the stations in the sequence.
Station Name	Name of the next station in the visitation sequence.
Step Name	A step name may optionally be associated with each station in the visitation sequence. The same step name may appear in different sequences, but not more than once in a given sequence.
Next Step	Name of the next step to be visited in the visitation sequence. If left blank, the default is for steps to be performed in the order in which they are defined in the sequence.
Assignments	Repeat group is used to define attribute, variable, picture, picture set, or other assignments.
Assignment Type	Type of assignment to be made before transferring the entity to the station specified in this step of the sequence. Assignment type may be **Attribute**, **Variable**, **Picture**, **Picture Set**, or **Other**.
Attribute Name	Name of the attribute to which the specified value will be assigned.
Variable Name	Name of the variable to which the specified value will be assigned.
Picture	Name of the picture that will be assigned to the entity's Entity.Picture value.
Picture Set	Name of the picture set for assigning an Entity.Picture value to an entity.
Set Index	Index into the picture set that will be used for assigning Entity.Picture.
Other	Defines any model status variable or special-purpose attribute to which the specified value will be assigned.
Value	Value to be assigned to the variable specified.

Conveyor module

DESCRIPTION

The Conveyor module allows the definition of either an accumulating or non-accumulating conveyor for entity movement between stations. When using a conveyor to transfer between modules, the conveyor must be defined using this module.

TYPICAL USES

- Baggage-handling belt
- Bottling conveyor

PROMPTS

Prompt	Description
Name	Name of the conveyor.
Segment Name	Name of the segment set defined in the segment module that is associated with the specified conveyor. The segment set defines the stations between which the conveyor transfers entities.
Type	Determines the conveyor type, either **Accumulating** or **Non-Accumulating**.
Velocity	The initial speed at which entities move along the conveyor in length units per base unit time.
Units	Velocity time units.
Cell Size	The smallest portion of a conveyor that an entity can occupy.
Max Cells Occupied	The maximum number of conveyor cells that any given entity will access at a time.
Accumulation Length	The accumulation length of the entity. This size is utilized in the simulation only when the conveyor type is Accumulating.
Initial Status	Initial status of the conveyor, either **Active** or **Inactive**.
Report Statistics	Specifies whether or not statistics will be collected automatically.

Segment module

DESCRIPTION

The Segment module defines the distance between two stations in the segment set of a conveyor. The beginning station, ending station, and segment are used to create the appropriate segment set, which defines the path of the conveyor. Typically, more than one Segment module is used to define the segment set.

TYPICAL USES

- Distillery operation that has a path from bottling to labeling to the boxing area
- Baggage-handling area that has a path from check-in to a holding area to a staging area

PROMPTS

Prompt	Description
Name	Name of the segment.
Beginning Station	The name of the first station that is located on the conveyor.
Next Station	The name of the next station that is located on the conveyor.
Length	Distance between this station and the previous station.

Transporter module

DESCRIPTION

The Transporter module allows the definition of a free-path or guided transporter device for entity movement from one location to another. Free-path transporters move freely between stations and are not influenced by other transporter traffic. The Distance module specifies the distances between stations. Guided transporters, on the other hand, are restricted to run on fixed paths such as tracks or rails. Movement may be affected by traffic congestion from other vehicles. Guided transporters travel on networks defined using the Network and Network Link modules.

Each spreadsheet entry creates a single transporter set.

When using a transporter to transfer between modules or as the server in a Leave or Enter module, the transporter must be defined using the Transporter module.

TYPICAL USES

- Forklifts that transfer parts between processing stations
- Wheelchairs that move patients between diagnostic areas

PROMPTS

Prompt	Description
Name	Name of the transporter. Can be used to represent a set of multiple transporter units; each unit within the set is capable of independent movement.
Number of Units	Number of independent transporter units in the transporter set.
Type	Specifies whether the transporter set is free path or guided.
Distance Set	Specifies the name of the distance set that contains the station names and distances between which the free path transporter will travel. Distances are defined in the Distance module.
Network Name	Specifies the name of the network that defines the distances and zone configuration of all paths that the guided transporter set may follow. Networks are defined in the Network module and the links/segments that make up a network are defined using the Network Link module.
Velocity	The initial velocity of the set of transporter units in length units per base time unit.
Units	Velocity time units.
Acceleration	Specifies the acceleration in length units per unit time squared that is applied to return a guided transporter to its normal operating velocity after slowing down. The unit time is specified in the Units field.
Deceleration	Specifies the deceleration in length units per unit time squared that is applied to brake a guided transporter when it needs to slow down or stop. The unit time is specified in the Units field.
Turning Velocity Factor	Defines the factor to be applied to a guided transporter's velocity when it negotiates a turn. In order for slow-downs through turns to be performed automatically, directions must be defined for all network links in the network Network Name.
Zone Control Rule	Determines when a guided transporter controlling a zone releases it, allowing another transporter access to that zone.
Units	This field specifies the time units for the Acceleration and Deceleration entries (i.e., per minute squared, per hour squared, etc.).
Initial Position Status	Repeat group that allows the specification of initial position. One entry may be used for each unit in the transporter set (as defined by Capacity).

Prompt	Description
Initial Position	Determines the type of initial position for the transporter unit(s). If a free-path transporter, Default will place the transporter at the first station location specified in the Distance module Distance Set. If a guided transporter, Default will place the transporter at the first network link specified in the Network module Network Name.
Station Name	Name of the station for the transporter's initial position.
Intersection Name	Defines the name of the intersection at which the guided transporter unit will be initialized.
Network Link Name	Defines the name of the network link at which the guided transporter unit will be initialized.
Zone	Defines the zone number on the network link at which the guided transporter unit will be initialized.
Initial Status	Establishes the initial status, **Active** or **Inactive**, of the transporter unit.
Size Based On	Determines whether the guided transporter's size will be specified in length units or number of zones.
Size Value	Specifies the initial transporter size in number of length units or zones according to the Size Based On field.
Report Statistics	Specifies whether or not statistics will be collected automatically.

Distance module

DESCRIPTION

The Distance module is used to define the travel distances between all stations that free-path transporters may visit. It consists of a distance name and a list of station pairs and individual distances.

TYPICAL USES

- Defining the distance a forklift travels between processing stations
- Defining the distance hospital gurneys travel between various diagnostic areas

PROMPTS

Prompt	Description
Name	Distance set name.
Beginning Station	Starting station name.
Ending Station	Ending station name.
Distance	Distance between the beginning and ending station.

Network module

DESCRIPTION

The Network module defines a system map that a set of guided transporters will follow. A network encompasses the set of links specified in its Network Links repeat group. The parameters of a network link (e.g., length, intersections, directions) are defined in the Network Link module.

TYPICAL USES

- Required when using guided vehicles

PROMPTS

Prompt	Description
Name	Name of the network.
Network Link Name	Name of network link included in the network.

Network Link module

DESCRIPTION

The Network Link module defines the characteristics of a guided transporter path between an intersection pair *Beginning Intersection Name* and *Ending Intersection Name*. The Network module then references a set of network links to define a network that guides transporters follow for movement.

TYPICAL USES

- Required when using guided vehicles

PROMPTS

Prompt	Description
Name	Name of the network link.
Type	The type of movement through the link.
Beginning Intersection Name	Name of beginning intersection of the intersection pair that the link connects.
Ending Intersection Name	Name of ending intersection of the intersection pair that the link connects.
Beginning Direction	Direction of link as it leaves the beginning intersection.
Ending Direction	Direction of link as it enters the ending intersection.
Number Of Zones	Number of zones in the link.
Zone Length	The length of each zone in the link. The total length of the link is the number of zones multiplied by the zone length.
Velocity Change Factor	Multiplier that is to be applied to the current velocity of any vehicle traveling through the link.

Activity Area module

DESCRIPTION

An Activity Area is an area that can be associated with one or more stations. These areas are used to collect statistics associated with a station. Each activity area may have an associated parent activity area that is used to define the hierarchy of the system modeled. Statistics are automatically rolled up through the activity area hierarchy.

TYPICAL USES

- A workstation
- A manufacturing line
- A manufacturing site
- The entire enterprise

PROMPTS

Prompt	Description
Name	Name of the activity area's parent.
Description	Optional field to describe the activity area.
Organization Level	Number used to define the hierarchy of the area.
Parent Activity Area	Name of the activity area's parent.
Report Statistics	Specifies whether or not statistics will automatically be collected and stored in the report database for this activity area.

6 The Flow Process Panel

To meet high-volume production requirements, manufacturing operations in the consumer packaged goods industry usually involve semi-continuous batch processing and/or high-speed packaging operations. While some of these operations can be modeled using discrete approximations, often a better approach is required.

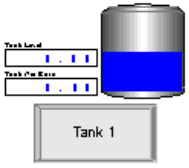
The Arena Flow Process template is one of a family of application solution templates (ASTs) built on the Arena simulation system. The flowchart and spreadsheet data modules in the Flow Process panel were designed specifically to model combined discrete-continuous systems (e.g., batch processing and material flow). If a system contains semi-continuous batch processes as well as high-speed packaging or filling operations, then it may be desirable to use modules from the Flow Process template with those from the Packaging template.

Flowchart modules

Flowchart modules are the set of objects that are placed in the model window to describe the simulation process.

Tank module

DESCRIPTION



The Tank module defines a location where material is held or stored.

The *Capacity* is the maximum quantity that may be stored in the tank. The *Initial Level* is the quantity in the tank at the beginning of the simulation or when the system is cleared.

The *Regulators* repeat group specifies a list of devices that may be used to add or remove material from the tank (e.g., valves or pumps) at a specified rate. Semi-continuous flow operations may then be executed by discrete entities using the regulators via the Flow module.

If a system contains both semi-continuous batch processes as well as high-speed packaging or filling operations, then it may be desirable to use the Flow Process panel with the Arena Packaging template. Enable the *Packaging Input Connection* and/or *Packaging Output Connection* options to connect a Flow Process Tank graphically to a Valve or Machine module in the Packaging panel.

When an Arena model (i.e., *.doe* file) is run, the simulation results are stored in a Microsoft Access database (*.mdb*) file by the same name (e.g., the simulation results for Banking.doe are stored in Banking.mdb). If statistics collection for all tanks is enabled in

the Project Parameters dialog AND Report Statistics is checked for an individual tank module, then Arena will automatically create, collect, and report statistics for that tank.

TYPICAL USES

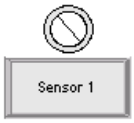
- Defines a location or holding area where bulk material is stored
- Defines a tank or vat holding liquids where the regulator represents a valve
- Defines a fluid-like material storage vessel where the regulator represents a gate valve
- Defines a bulk storage area or pile where the regulator represents a piler/reclaimer

PROMPTS

Prompt	Description
Name	The tank's identifier.
Capacity	The tank's capacity.
Initial Level	The quantity in the tank at the start of the simulation.
Regulator Name	Specifies a device that is used to add to or remove material from the tank (e.g., a valve name).
Maximum Rate	The initial maximum rate for the regulator.
Units	Time units of maximum rate.
Packaging Input Connection	Specifies whether the tank is connected to and allows input flow from a Valve module in the Packaging panel.
Packaging Output Connection	Specifies whether the tank is connected to and allows output flow to a Valve or Machine module in the Packaging panel. The Machine Type for the Machine module must be Filling Machine.
Animation	Specifies whether to display default animation of the tank's level and net rate.
Report Statistics	Specifies whether to report statistics on the tank.

Sensor module

DESCRIPTION



The Sensor module defines a detection device that monitors the level of material in a tank (Tank module). A sensor's location is specified using the *Tank Name*, *Location Type*, *Level*, or *Percentage* prompts.

A sensor is activated when its location is crossed in the *Crossing Direction* AND the sensor is enabled. When the sensor is activated, one or more *Actions* may be executed. Additionally, the *Create Discrete Entity* option may be used to create a discrete entity and send it to custom logic.

The Initial State field specifies whether the sensor is enabled or disabled at the beginning of the simulation run. If a sensor is disabled, it will be ignored and never activate. A sensor may be enabled/disabled dynamically at any point during the simulation run by assigning the SensorState (*Sensor ID*) variable.

TYPICAL USES

- Defines a device to monitor tank level so decisions can be made based on sensor state
- Defines a device to initiate action when a tank becomes full, empty, or some other significant level is crossed
- Defines a device to trigger re-ordering of material when stock runs low
- Defines a device to start and stop reclamations from a pile based on inventory level

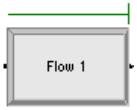
PROMPTS

Prompt	Description
Name	The sensor's identifier.
Tank Name	The name of the tank where the sensor is located.
Level	The level location of the sensor if Location Type is *Specific Level*.
Percentage (0-100)	The percentage location of the sensor if Location Type is *Percentage Capacity*.
Crossing Direction	The sensor is activated (triggered) when its location is crossed in this direction.
Initial State	Specifies whether the sensor is enabled or disabled at the beginning of the simulation run.
If Sensor is Activated:	
Action Type	Specifies the type of action to perform.
Regulator Name	The name of the regulator to regulate.

Prompt	Description
New Maximum Rate	The new maximum rate for the regulator
Units	Time units of new maximum rate.
Signal Value	The signal value to send.
Variable Name	The name of the variable to assign.
New Value	The new variable value.
Create Discrete Entity	Specifies whether to create a discrete entity when the sensor is activated (triggered).
Animation	Specifies whether to display a "covered/uncovered" global picture for the sensor.

Flow module

DESCRIPTION



The Flow module is used in conjunction with the Tank module to model semi-continuous flow operations such as adding material to a tank, removing material from a tank, or transferring material between two tanks.

When an entity enters the Flow module, a flow operation of the specified Type is initiated (i.e., Add, Transfer, or Remove). The tanks affected by the flow are determined by the regulators specified as the source and/or destination. A regulator is a device (e.g., a valve) that is used to add or remove material from a tank. A tank's regulators are defined in the Tank module.

The entity is held in the Flow module until its flow operation is completed. The flow operation is completed when the *first* of three possible conditions is satisfied:

- When the specified *Quantity* has been transferred.
- When the specified *Time* has elapsed.
- When the specified *Signal* Value is received via a Signal module.

The flow rate of the entity is regulated according to the maximum rate(s) of the source regulator and/or destination regulator. Additionally, the flow rate may be further constrained due to starvation or blocking from empty or full tanks. The initial maximum rate of a regulator is specified in the Tank module. A regulator's maximum rate may be changed during a run using the Regulate module. The RegulatorMaxRate(*Regulator ID*) function returns the current maximum rate of a regulator. The RegulatorRate(*Regulator ID*) function returns the current rate of a regulator.

The actual quantity transferred by an entity in the Flow module may be stored in the *Quantity Save Attribute* when the entity exits the module.

TYPICAL USES

- Adding material to a tank
- Removing material from a tank
- Transferring material between two tanks

PROMPTS

Prompt	Description
Name	The module's identifier.
Type	The flow type, which may be **Transfer**, **Add**, or **Remove**.
Flow Destination/Source:	
Source Regulator Type	Indicates how the source regulator will be specified; as **Regulator**, **Regulator Set**, **Attribute**, **Expression**.
Regulator Name	Name of the source regulator.
Regulator Set Name	Name of the regulator set that contains the source regulator name.
Set Index	Index into the regulator set.
Attribute Name	Name of the attribute that stores the source regulator name.
Expression	Expression that evaluates to the source regulator name.
Destination Regulator Type	Indicates how the destination regulator will be specified; as **Regulator**, **Regulator Set**, **Attribute**, **Expression**.
Regulator Name	Name of the destination regulator
Regulator Set Name	Name of the regulator set that contains the destination regulator name.
Set Index	Index into the regulator set.
Attribute Name	Name of the attribute that stores the destination regulator name.
Expression	Expression that evaluates to the destination regulator name.
Stop Flow After:	
Quantity	The flow will be stopped after this quantity has been transferred.
Time	The flow will be stopped after this time duration has elapsed.
Units	The time units of the *Time*.
Signal Value	The flow will be stopped if this signal value is received.

Prompt	Description
Priority	The priority of the flow operation when allocating limited input or output to multiple flow operations occurring at a tank.
Allocation	Determines how the time and costs accrued during the flow operation will be allocated to the entity.
Quantity Save Attribute	Attribute to save the quantity transferred by the entity.

Regulate module

DESCRIPTION

The Regulate module provides discrete control for assigning the maximum rates of regulators defined in the Tank module.

TYPICAL USES

- Used to simulate such actions as the closing or opening of valves

PROMPTS

Prompt	Description
Name	The module description.
Maximum Rate Assignments:	
Regulator Type	Indicates how the regulator will be specified; defined as **Regulator**, **Regulator Set**, **Attribute**, **Expression**.
Regulator Name	Name of the regulator.
Regulator Set Name	Name of the regulator set that contains the regulator.
Set Index	Index into the regulator set.
Attribute Name	Name of the attribute that stores the regulator name.
Expression	Expression that evaluates to the regulator name.
New Maximum Rate	The new maximum rate.
Units	Time units of new maximum rate.

Seize Regulator module

DESCRIPTION

A tank regulator may be used for only one flow operation at any given time.

The Seize Regulator module allocates one or more regulators to an entity. When an entity enters this module, it waits in a queue until all specified regulators are available simultaneously. An allocated regulator is released by an entity using the Release Regulator module.

TYPICAL USES

- Controls ownership of regulators and avoid multiple entities trying to use the same regulator in a Flow module simultaneously
- Chooses from a set of alternative regulators using a selection rule

PROMPTS

Prompt	Description
Name	The module identifier.
Priority	Priority value of the entity waiting at this module for the regulator(s) specified if one or more entities from other Seize Regulator modules are waiting for the same regulator(s).
Regulators:	
Regulator Type	Indicates how the regulator to seize will be specified; as **Regulator**, **Regulator Set**, **Attribute**, or **Expression**.
Regulator Name	Name of the regulator to seize.
Regulator Set Name	Name of the regulator set that contains the regulator to seize.
Selection Rule	Method of selecting among available resources in a set. **Cyclical** will cycle through available members (1-2-3-1-2-3). **Random** will randomly select a member. **Preferred Order** will always select the first available member as listed in the set (1 if available, then 2 if available, then 3). **Specific Member** requires an index value to specify which member of the set. **User Rule** may be used to code custom selection rules. The number argument is passed to the UserRule routine (which may be coded in either VBA or a user-coded dll).
Save Attribute	Attribute name used to store the index number into the set of the member that was selected. This attribute can then later be referenced in Flow, Regulate, and Release Regulator modules to identify the regulator that was chosen and seized.
Set Index	Index into the regulator set.

Prompt	Description
Rule Number	Specifies number argument that is passed to the UserRule routine.
Attribute Name	Name of the attribute that stores the name of the regulator to seize.
Expression	Expression that evaluates to the name of the regulator to seize.

Release Regulator module

DESCRIPTION

The Release Regulator module is used to release tank regulators that have been allocated to an entity using the Seize Regulator module. This makes those regulators available to other entities waiting to seize the regulator(s).

When the entity enters the Release Regulator module, it gives up control of the specified regulator(s). Any entities waiting in queues for those regulators will gain control of the regulators immediately.

TYPICAL USES

- Releases control of a regulator. A regulator is a device that adds or removes material from tank (e.g., a valve).

PROMPTS

Prompt	Description
Name	The module identifier.
Regulators:	
Regulator Type	Indicates how the regulator to release will be specified; defined as **Regulator**, **Regulator Set**, **Attribute**, **Expression**.
Regulator Name	Name of the regulator to release.
Regulator Set Name	Name of the regulator set that contains the regulator to release.
Release Rule	Method of determining which regulator within a set to release. **Last Member Seized** and **First Member Seized** will release the last/first member from within the set that was seized.**Specific Member** indicates that a member number or attribute (with a member number value) will be used to specify the member to release.
Set Index	Index into the regulator set.

Prompt	Description
Attribute Name	Name of the attribute that stores the name of the regulator to release.
Expression	Expression that evaluates to the name of the regulator to release.

Data module

Data modules are the set of objects in the spreadsheet view of the model that define the characteristics of various process elements, such as resources and queues.

Regulator Set module

DESCRIPTION

The Regulator Set module defines a set of regulators to select from and use in other Flow Process modules (Seize Regulator, Release Regulator, Regulate, Flow).

TYPICAL USES

- Defines a grouping of regulators for logical purposes (e.g., to select an available member from the set using a selection rule)

PROMPTS

Prompt	Description
Name	The set's identifier.
Regulator Name	Regulator member in set.

Statistical Distributions

Arena contains a set of built-in functions for generating random numbers from the commonly used probability distributions. These distributions appear on pull-down menus in many Arena modules where they're likely to be used. They also match the distributions in the Arena Input Analyzer. This appendix describes all of the Arena distributions.

Each of the distributions in Arena has one or more parameter values associated with it. You must specify these parameter values to define the distribution fully. The number, meaning, and order of the parameter values depend on the distribution. A summary of the distributions (in alphabetical order) and parameter values is given in the table below.

Summary of Arena's Probability Distributions

Distribution		Parameter Values
Beta	BETA	Beta, Alpha
Continuous	CONT	$CumP_1$,Val_1, . . . $CumP_n$,Val_n
Discrete	DISC	$CumP_1$,Val_1, . . . $CumP_n$,Val_n
Erlang	ERLA	ExpoMean, k
Exponential	EXPO	Mean
Gamma	GAMM	Beta, Alpha
Johnson	JOHN	Gamma, Delta, Lambda, Xi
Lognormal	LOGN	LogMean, LogStd
Normal	NORM	Mean, StdDev
Poisson	POIS	Mean
Triangular	TRIA	Min, Mode, Max
Uniform	UNIF	Min, Max
Weibull	WEIB	Beta, Alpha

To enter a distribution in an Arena field, you type the name of the distribution (or its four-letter abbreviation) followed by its parameters enclosed in parentheses. You may use spaces around punctuation to help read the distribution. A few examples appear below.

UNIF(3.5, 6)

Uniform distribution with a minimum value of 3.5, a maximum value of 6

NORMAL(83, 12.8)

Normal distribution with a mean of 83, a standard deviation of 12.8

DISCRETE(0.3,50, 0.75,80, 1.0,100)

Discrete probability distribution that will return a value of 50 with probability 0.3, a value of 80 with cumulative probability 0.75, and a value of 100 with cumulative probability of 1.0. (See "Discrete Probability" for a description of these parameters.)

TRIA(10, 15, 22)

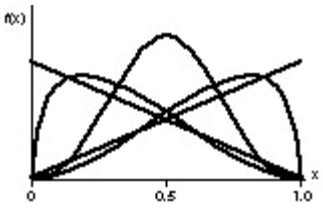
Triangular distribution with a minimum value of 10, mode (most likely value) of 15, and maximum value of 22

In the following pages, we provide a summary of each of the distributions supported by Arena, listed in alphabetical order for easy reference. Each includes the density or mass function, parameters, range, mean, variance, and typical applications for the distribution.

If you have existing data and want to select the appropriate distribution for use in your model, use Arena's Input Analyzer. Click on **Tools > Input Analyzer** to launch the program, or launch it from the Windows Start menu.

Beta(β, α) BETA(Beta, Alpha)

Probability Density Function



$$f(x) = \begin{cases} \dfrac{x^{\beta-1}(1-x)^{\alpha-1}}{B(\beta,\alpha)} & \text{for } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

where β is the complete beta function given by

$$B(\beta, \alpha) = \int_0^1 t^{\beta-1}(1-t)^{\alpha-1} dt$$

Parameters Shape parameters Beta (β) and Alpha (α) specified as positive real numbers.

Range [0, 1] (Can also be transformed to [*a,b*] as described below)

Mean

$$\frac{\beta}{\beta + \alpha}$$

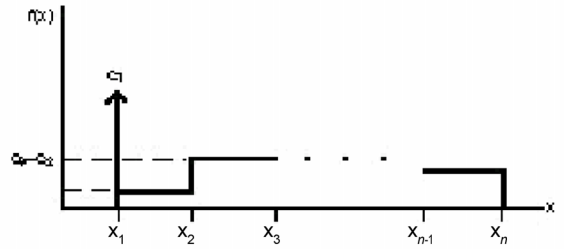
Variance

$$\frac{\beta a}{(\beta + a)^2(\beta + a + 1)}$$

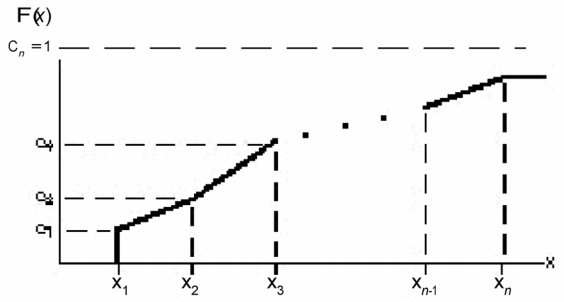
Applications Because of its ability to take on a wide variety of shapes, this distribution is often used as a rough model in the absence of data. Also, because the range of the beta distribution is from 0 to 1, the sample *X* can be transformed to the scaled beta sample *Y* with the range from *a* to *b* by using the equation $Y = a + (b - a)X$. The beta is often used to represent random proportions, such as the proportion of defective items in a lot.

Continuous **CONTINUOUS(CumP$_1$, Val$_1$, . . ., CumP$_n$, Val$_n$)**
(c_1, x_1,. . ., c_n,x_n)

Probability Density Function



Cumulative Distribution Function



$$f(x) = \begin{cases} c_1 & \text{if } x = x_1 \text{ (a mass of probability } c_1 \text{ at } x_1) \\ c_j - c_{j-1} & \text{if } x_{j-1} \le x < x_j, \text{ for } j = 2, 3, \ldots, n \\ 0 & \text{if } x < x_1 \text{ or } x \ge x_n \end{cases}$$

Parameters

The CONTINUOUS function in Arena returns a sample from a user-defined distribution. Pairs of cumulative probabilities c_j (= CumP$_j$) and associated values x_j (= Val$_j$) are specified. The sample returned will be a real number between x_1 and x_n, and will be less than or equal to each x_j with corresponding cumulative probability c_j. The x_j's must increase with j. The c_j's must all be between 0 and 1, must increase with j, and c_n must be 1.

The cumulative distribution function $F(x)$ is piecewise linear with "corners" defined by $F(x_j) = c_j$ for $j = 1, 2, \ldots, n$. Thus, for $j \ge 2$, the returned value will be in the interval $(x_{j-1}, x_j]$

with probability $c_j - c_{j-1}$; given that it is in this interval, it will be distributed uniformly over it.

You must take care to specify c_1 and x_1 to get the effect you want at the left edge of the distribution. The CONTINUOUS function will return (exactly) the value x_1 with probability c_1. Thus, if you specify $c_1 > 0$, this actually results in a mixed discrete-continuous distribution returning (exactly) x_1 with probability c_1, and with probability $1 - c_1$ a continuous random variate on $(x_1, x_n]$ as described above. The graph of $F(x)$ above depicts a situation where $c_1 > 0$. On the other hand, if you specify $c_1 = 0$, you will get a (truly) continuous distribution on $[x_1, x_n]$ as described above, with no "mass" of probability at x_1; in this case, the graph of $F(x)$ would be continuous, with no jump at x_1.

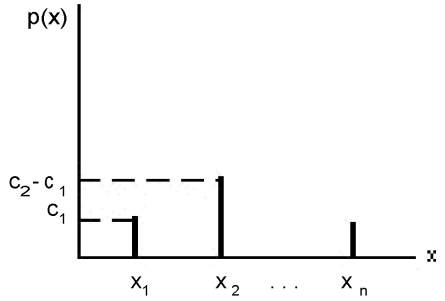
As an example use of the CONTINUOUS function, suppose you have collected a set of data $x_1, x_2, \ldots, x_n$ (assumed to be sorted into increasing order) on service times, for example. Rather than using a fitted theoretical distribution from the Input Analyzer, you want to generate service times in the simulation "directly" from the data, consistent with how they're spread out and bunched up, and between the minimum x_1 and the maximum x_n you observed. Assuming that you don't want a "mass" of probability sitting directly on x_1, you'd specify $c_1 = 0$ and then $c_j = (j - 1)/(n - 1)$ for $j = 2, 3, \ldots, n$.

Range $[x_1, x_n]$

Applications The continuous empirical distribution is often used to incorporate actual data for continuous random variables directly into the model. This distribution can be used as an alternative to a theoretical distribution that has been fitted to the data, such as in data that have a multimodal profile or where there are significant outliers.

Discrete **DISCRETE(CumP$_1$, Val$_1$, . . ., CumP$_n$, Val$_n$)**
(c_1, x_1, . . ., c_n, x_n)

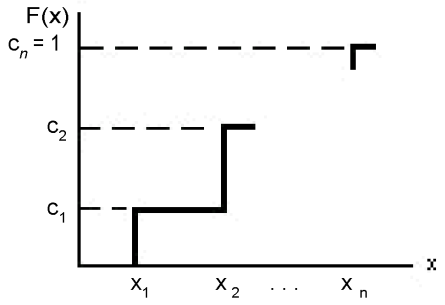
Probability Mass Function



$p(x_j) = c_j - c_{j-1}$

where $c_0 = 0$

Cumulative Distribution Function



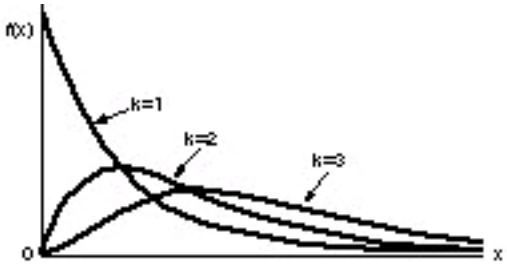
Parameters The DISCRETE function in Arena returns a sample from a user-defined discrete probability distribution. The distribution is defined by the set of n possible discrete values (denoted by $x_1, x_2, \ldots, x_n$) that can be returned by the function and the cumulative probabilities (denoted by $c_1, c_2, \ldots, c_n$) associated with these discrete values. The cumulative probability (c_j) for x_j is defined as the probability of obtaining a value that is less than or equal to x_j. Hence, c_j is equal to the sum of $p(x_k)$ for k going from 1 to j. By definition, $c_n = 1$.

Range $\{x_1, x_2, \ldots, x_n\}$

Applications The discrete empirical distribution is often used to incorporate discrete empirical data directly into the model. This distribution is frequently used for discrete assignments such as the job type, the visitation sequence, or the batch size for an arriving entity.

Erlang(β, k) ERLANG(ExpMean, k) or ERLA(ExpMean, k)

Probability Density Function



$$f(x) = \begin{cases} \dfrac{\beta^{-k} x^{k-1} e^{-x/\beta}}{(k-1)!} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Parameters If X1, X2, . . . , Xk are independent, identically distributed exponential random variables, then the sum of these k samples has an Erlang-k distribution. The mean (β) of each of the component exponential distributions and the number of exponential random variables (k) are the parameters of the distribution. The exponential mean is specified as a positive real number, and k is specified as a positive integer.

Range $[0, +\infty)$

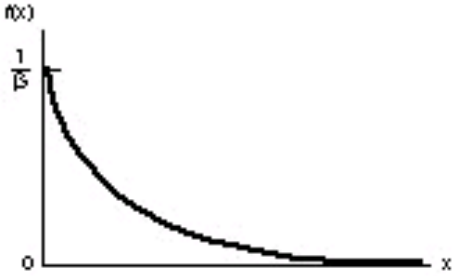
Mean $k\beta$

Variance $k\beta^2$

Applications The Erlang distribution is used in situations in which an activity occurs in successive phases and each phase has an exponential distribution. For large k, the Erlang approaches the normal distribution. The Erlang distribution is often used to represent the time required to complete a task. The Erlang distribution is a special case of the gamma distribution in which the shape parameter, α, is an integer (k).

Exponential(β) EXPONENTIAL(Mean) or EXPO(Mean)

Probability Density Function



$$f(x) = \begin{cases} \frac{1}{\beta} e^{-x/\beta} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Parameters The mean (β) specified as a positive real number.

Range $[0, +\infty)$

Mean β

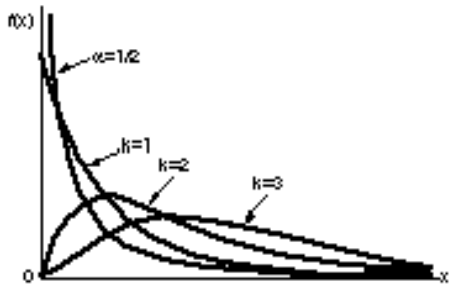
Variance β^2

Applications This distribution is often used to model inter-event times in random arrival and breakdown processes, but it is generally inappropriate for modeling process delay times.

In Arena's Create module, the Schedule option automatically samples from an exponential distribution with a mean that changes according to the defined schedule. This is particularly useful in service applications, such as retail business or call centers, where the volume of customers changes throughout the day.

Gamma(β, α) GAMMA(Beta, Alpha) or GAMM(Beta, Alpha)

Probability Density Function



$$f(x) = \begin{cases} \dfrac{\beta^{-\alpha} x^{\alpha-1} e^{-x/\beta}}{\Gamma(\alpha)} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

where Γ is the complete gamma function given by

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-1} dt$$

Parameters Shape parameter (α) and scale parameter (β) specified as positive real values.

Range $[0, +\infty)$

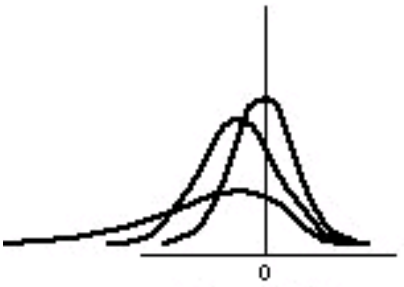
Mean $\alpha\beta$

Variance $\alpha\beta^2$

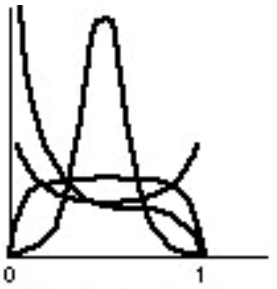
Applications For integer shape parameters, the gamma is the same as the Erlang distribution. The gamma is often used to represent the time required to complete some task (e.g., a machining time or machine repair time).

Johnson JOHNSON(Gamma, Delta, Lambda, Xi) or JOHN(Gamma, Delta, Lambda, Xi)

Probability Density Function



Unbounded Family



Bounded Family

Parameters Gamma shape parameter (γ), Delta shape parameter ($\delta > 0$), Lambda scale parameter (λ), and Xi location parameter (ξ).

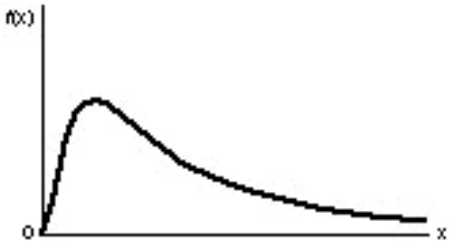
Range

$(-\infty, +\infty)$	Unbounded Family
$[\xi, \xi + \lambda]$	Bounded Family

Applications The flexibility of the Johnson distribution allows it to fit many data sets. Arena can sample from both the unbounded and bounded form of the distribution. If Delta (δ) is passed as a positive number, the bounded form is used. If Delta is passed as a negative value, the unbounded form is used with $|\delta|$ as the parameter.

Lognormal(μ, σ) LOGNORMAL(LogMean, LogStd) or LOGN(LogMean, LogStd)

Probability Density Function



Denote the user-specified input parameters as LogMean = μ_l and LogStd = σ_l.

Then let $\mu = \ln(\mu_l^2 / \sqrt{\sigma_l^2 + \mu_l^2})$ and

$$\sigma = \sqrt{\ln\left[(\sigma_l^2 + \mu_l^2)/\mu_l^2\right]}$$

the probability density function can then be written as

$$f(x) = \begin{cases} \dfrac{1}{\sigma x\sqrt{2\pi}} e^{-(\ln(x)-\mu)^2/(2\sigma^2)} & \text{for } x > 0 \\ 0 \quad \text{otherwise} \end{cases}$$

Parameters Mean LogMean ($\mu_1 > 0$) and standard deviation LogStd ($\sigma_1 > 0$) of the lognormal random variable. Both LogMean and LogStd must be specified as strictly positive real numbers.

Range $[0, +\infty)$

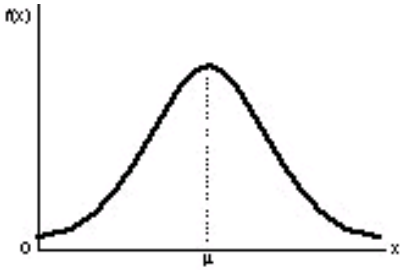
Mean LogMean = $\mu_1 = e^{\mu + \sigma^2/2}$

Variance (LogStd)$^2 = \sigma_1^2 = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$

Applications The lognormal distribution is used in situations in which the quantity is the product of a large number of random quantities. It is also frequently used to represent task times that have a distribution skewed to the right. This distribution is related to the normal distribution as follows. If X has a lognormal (μ_l, σ_l) distribution, then ln(X) has a normal (μ, σ) distribution. Note that μ and σ are *not* the mean and standard deviation of the lognormal random variable X, but rather the mean and standard deviation of the normal random variable lnX.

Normal(μ, σ) NORMAL(Mean, StdDev) or NORM(Mean, StdDev)

Probability Density Function



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)} \text{ for all real } x$$

Parameters The mean (μ) specified as a real number and standard deviation (σ) specified as a positive real number.

Range $(-\infty, +\infty)$

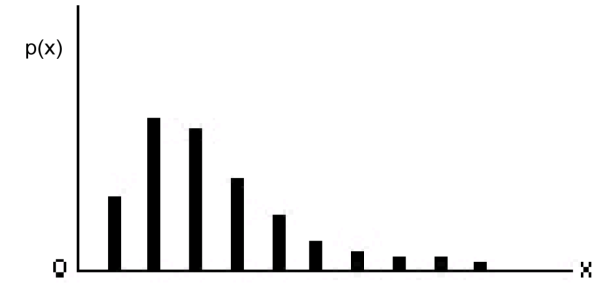
Mean μ

Variance σ^2

Applications The normal distribution is used in situations in which the central limit theorem applies; i.e., quantities that are sums of other quantities. It is also used empirically for many processes that appear to have a symmetric distribution. Because the theoretical range is from $-\infty$ to $+\infty$, the distribution should only be used for positive quantities like processing times.

Poisson(λ) POISSON(Mean) or POIS(Mean)

Probability Mass Function



$$p(x) = \begin{cases} \dfrac{e^{-\lambda}\lambda^{x}}{x!} & \text{for } x \varepsilon \{0, 1, ...\} \\ 0 & \text{otherwise} \end{cases}$$

Parameters The mean (λ) specified as a positive real number.

Range $\{0, 1, \ldots\}$

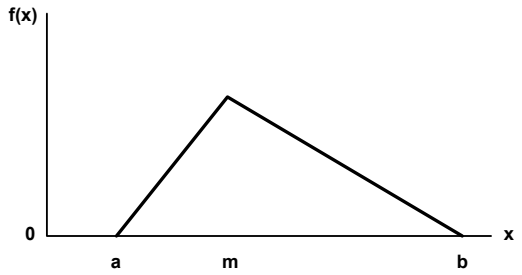
Mean λ

Variance λ

Applications The Poisson distribution is a discrete distribution that is often used to model the number of random events occurring in a fixed interval of time. If the time between successive events is exponentially distributed, then the number of events that occur in a fixed-time interval has a Poisson distribution. The Poisson distribution is also used to model random batch sizes.

Triangular(*a, m, b*) TRIANGULAR(Min, Mode, Max) or TRIA(Min, Mode, Max)

Probability Density Function



$$f(x) = \begin{cases} \dfrac{2(x-a)}{(m-a)(b-a)} & \text{for } a \le x \le m \\ \dfrac{2(b-x)}{(b-m)(b-a)} & \text{for } m \le x \le b \\ 0 & \text{otherwise} \end{cases}$$

Parameters The minimum (a), mode (m), and maximum (b) values for the distribution specified as real numbers with $a < m < b$.

Range $[a, b]$

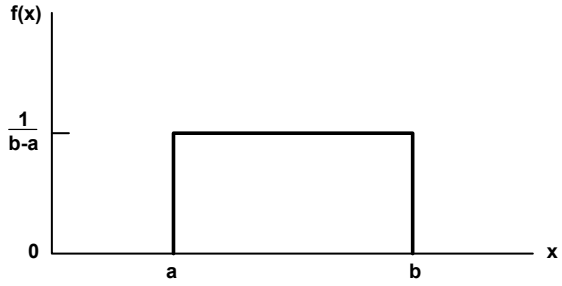
Mean $(a + m + b)/3$

Variance $(a^2 + m^2 + b^2 - ma - ab - mb)/18$

Applications The triangular distribution is commonly used in situations in which the exact form of the distribution is not known, but estimates (or guesses) for the minimum, maximum, and most likely values are available. The triangular distribution is easier to use and explain than other distributions that may be used in this situation (e.g., the beta distribution).

Uniform(*a, b*) UNIFORM(Min, Max) or UNIF(Min, Max)

Probability Density Function



$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \le x \le b \\ 0 & \text{otherwise} \end{cases}$$

Parameters The minimum (a) and maximum (b) values for the distribution specified as real numbers with $a < b$.

Range $[a, b]$

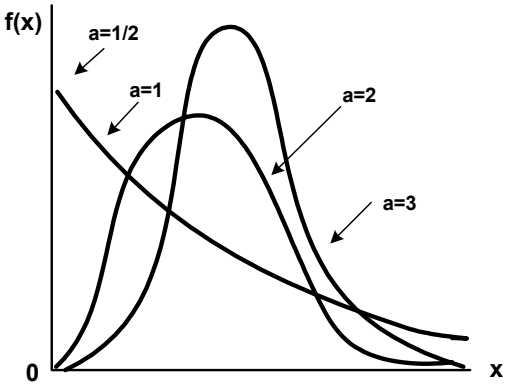
Mean $(a + b)/2$

Variance $(b - a)^2/12$

Applications The uniform distribution is used when all values over a finite range are considered to be equally likely. It is sometimes used when no information other than the range is available. The uniform distribution has a larger variance than other distributions that are used when information is lacking (e.g., the triangular distribution).

Weibull(β, α) WEIBULL(Beta, Alpha) or WEIB(Beta, Alpha)

Probability Density Function



$$f(x) = \begin{cases} a\beta^{-a}x^{a-1}e^{-(x/\beta)^a} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Parameters Shape parameter (α) and scale parameter (β) specified as positive real values.

Range $[0, +\infty)$

Mean $\frac{\beta}{a}\Gamma\left(\frac{1}{a}\right)$, where Γ is the complete gamma function (see gamma distribution).

Variance $\frac{\beta^2}{a}\left\{2\Gamma\left(\frac{2}{a}\right) - \frac{1}{a}\left[\Gamma\left(\frac{1}{a}\right)\right]^2\right\}$

Applications The Weibull distribution is widely used in reliability models to represent the lifetime of a device. If a system consists of a large number of parts that fail independently, and if the system fails when any single part fails, then the time between successive failures can be approximated by the Weibull distribution. This distribution is also used to represent non-negative task times that are skewed to the left.

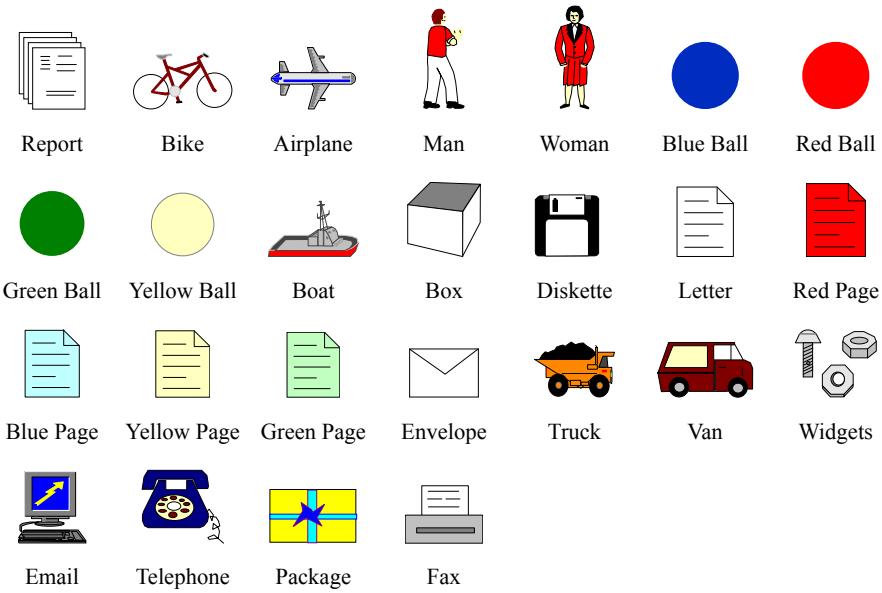
B Library Shapes and Symbols

Arena *.plb* picture files

Arena's picture libraries (**.plb* files) contain images that you may use for entity, resource, and global pictures. You can use the images in these libraries for any of these three types of picture animation. You also can copy these pictures into your own libraries, draw new pictures in Arena's picture editor, or add pictures from the Arena Symbol Factory to any *.plb* file. View the online help topic "Animating Objects" to learn more about adding symbols to your picture libraries.

The Basic Process library supplies a set of default entity pictures that are preloaded into your Arena models. When you start a model, Arena adds the pictures from *BasicProcess.plb* to the entity picture list (see the **Edit > Entity Pictures** menu). Also, the predefined names associated with these pictures are added to the entity picture list (e.g., *Picture.Report*) in the Entity module and Entity picture placement dialog.

Example pictures in Basic Process.plb



Arena .plb libraries

The list of standard Arena picture libraries includes:

- Buldings.plb
- Equipment.plb
- Faces.plb
- Factory.plb
- General.plb
- Machines.plb
- Medical.plb
- Objects.plb
- Office.plb
- Office Equipment.plb
- People.plb
- Symbols.plb
- Vehicles.plb
- Workers.plb

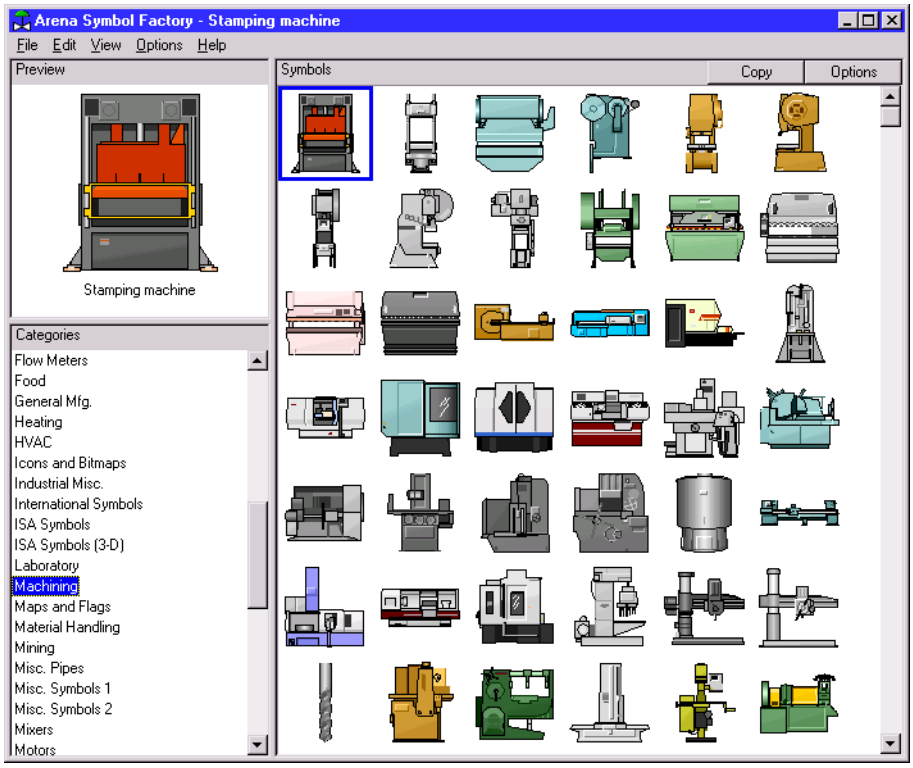
Common shapes may be placed from any of these categories, or they may be added from the Arena Symbol Factory or other image source by pasting from your Clipboard into a specific library. If you wish to create your own library of shapes, simply select **New** from the Entity Picture Placement window. The "Current Library" displays "(None Opened)" because the new library has not yet been named or saved. Simply select **Add** for each library shape you wish to include in your new library file (or an existing one), and draw or paste the appropriate images onto the picture editor board. Remember to drag-and-drop the Ride Point to the appropriate position on your image.

Arena Symbol Factory

Arena animations can also be enhanced using Arena Symbol Factory's extensive library of over 4,000 symbols. These symbols can be used for entity, resource, transporter, or global pictures, or as graphic symbols within a model window. You can copy these images directly to the Arena model window, add them to your own libraries (.plb files), or add them to any of the standard Arena picture library files.

The Symbol Factory application

The Symbol Factory is launched from the main Arena toolbar by selecting **Tools > Arena Symbol Factory**. The application screen contains three distinct areas: the *Preview* pane, showing an enlarged image of the currently selected symbol; the *Categories* pane, containing the list of library symbols by groupings; and the *Symbol* pane, displaying images of all shapes contained within a selected category. An example of the application is shown below with the Machining category selected.



The basic operation of Symbol Factory simply requires you to select an object from a category list, click **Edit > Copy**, and then paste the symbol into your picture editor window or the Arena application window (for static shapes). Details on the Arena Symbol Factory can be found in the extensive online help embedded in the application.

You must have suitable license activation to use the full Symbol Factory capabilities. In Arena training/evaluation, Basic Edition, and run-time versions, Arena Symbol Factory is only available in demo mode. Demo mode limits use to only the first symbol in each category. New categories cannot be created, and symbols cannot be moved, added, deleted, imported, or exported in demo mode.

Using Visio and Arena

Introduction

Since the mid-1980s, Business Process Improvement (BPI) and simulation have become leading methodologies to allow corporations to deliver high-quality products and services through business process modeling. Businesses are seeking not simply to automate existing operations, but to improve and redesign business processes to meet and exceed customers' expectations for the delivery of products and services.

Many simulation systems are standalone, running only on a desktop and having limited capabilities to integrate with other technologies such as company databases, workflow engines, and other desktop solutions. At the same time, leading corporations have discovered the value of simulation technology. Many companies require simulation to be used as standard procedure in support of process improvements.

A much greater return on investment is possible if process descriptions and simulation results can be accessed throughout the enterprise. Microsoft® Visio®, the leading desktop drawing product, supplies a common graphics platform for representing business process models. Visio's open architecture via ActiveX® and Visual Basic® for Applications enables the overall modeling tool to be customized to fit changing needs. The models also can be shared throughout the enterprise by publishing on an intranet directly from Visio.

Arena software includes a built-in interface to Visio, mapping shapes from a Visio drawing to the appropriate simulation constructs in Arena. Process hierarchy is captured in the Arena simulation model, including activity-based costing and process logic, enabling an understanding of the distribution of cost and effort among process components. As with Visio, Arena supplies robust ActiveX support, a Microsoft® Office-compatible interface, and VBA to fulfill enterprise needs for best-of-breed software products.

The Visio Process Simulation add-in

If you use Visio, you can immediately put Arena's process mapping tool to work with the Visio Process Simulator. Created as an add-in to Visio, the Process Simulator includes a Visio template and stencil for mapping processes with shapes designed to mirror the flowchart and data modules in the Basic Process panel of Arena. When you've completed your Visio process map using these tools, you're ready to simulate it in Arena!

Start a drawing

To begin a drawing, browse to the Process Simulation template, which the Process Simulator installs in a Process Simulation folder under Visio's Solutions area. This dialog

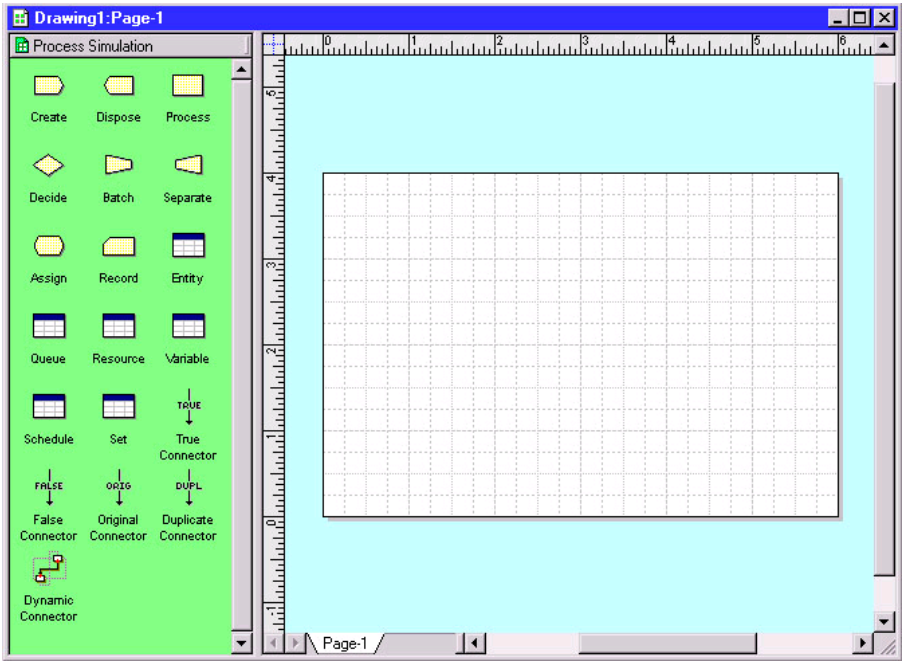
box typically appears when you first start Visio, or you can bring it up via the **File > New > Choose Drawing Type > Browse Templates** menu.

It's important to start your drawing by browsing to the Process Simulation template, rather than opening or starting a new drawing and directly attaching the Process Simulation stencil. If you don't start your drawing with the template, custom dialogs and other features may not be available.

The Process Simulation template sets up the Visio environment to work with Arena and opens a new drawing. Drawings that are started with this template will automatically display the Process Simulation stencil, provide custom dialog boxes for editing shape properties, and add a Simulate menu to Visio for checking your drawing and transferring it to Arena.

The Process Simulation stencil, which is opened with your new drawing, contains shapes that will capture your process flow and data. The shapes mirror those in Arena's Basic Process panel, so you can use Visio to define fully a model of your process, including all of the parameters needed to exploit the power of simulation analysis.

The first 14 shapes in the Process Simulation stencil correspond to the Arena flowchart and data modules with the same names. In addition, the Visio stencil contains five connector shapes for properly defining the process flowchart.



Place and connect shapes

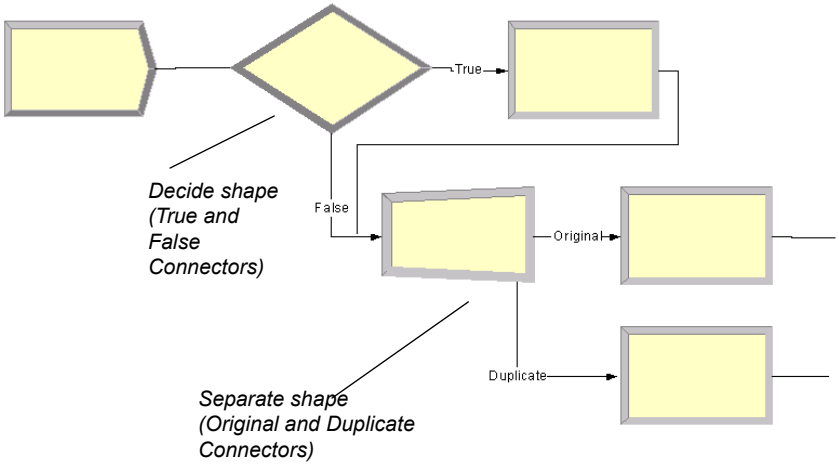
You can change the Visio page layout or size via the File > Page Setup menu. For many process maps, a Standard page size and Landscape orientation work

To define the process flow portion of your model, drag and drop flowchart shapes (e.g., Create, Process, Dispose) from the Process Simulation stencil into your drawing, just as you would with other standard Visio stencils. Then, use the connectors from the Process Simulation stencil to establish flow.

When connecting shapes, the Dynamic Connector is used for all except Decide and Separate, which use special connectors to determine the type of each of the two possible outgoing connections. The table below lists the shapes from which the connectors can start (e.g., the True Connector can only start at a Decide shape).

Connector	Valid Starting Shape
True Connector	Decide
False Connector	Decide
Original Connector	Separate
Duplicate Connector	Separate
Dynamic Connector	Create, Process, Batch, Assign, Record

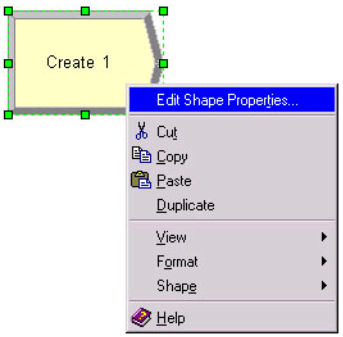
The sample flowchart below illustrates the use of these connectors, including the Dynamic Connector, which is unlabeled.



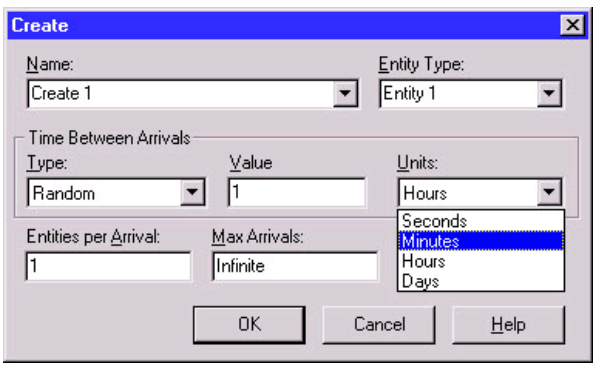
Enter custom properties

When you place shapes from the Process Simulation stencil, default names appear (e.g., Create 1, Process 1) to identify the steps in the process flow. Typically, you'll replace these names with descriptions of the process steps. To do so, simply select the shape and type the new name, as with other standard Visio shapes.

The Process Simulation stencil's shapes also provide custom dialogs for defining property values, including lists of valid options and other features to aid you in quickly characterizing process data. To open the dialog box, either double-click on the shape or right-click and select **Edit Shape Properties** from the pop-up menu, as shown below.



Either of these actions will display the appropriate dialog box for the shape. The dialogs that Arena presents for its corresponding modules are designed to match those in Visio, so that you're working in a familiar interface, whether mapping your process in Visio or Arena.



The property values that you define via the custom dialog boxes are stored with the individual shapes. When you transfer the process map to Arena, these parameters are used for simulation.

Add data shapes

To define the data modules (e.g., Entity, Resource) for the model of your process, you place shapes directly in the drawing for each of the elements you want to define. When the individual data shapes are transferred to Arena, they will not appear in the model window, but will be added to the appropriate spreadsheets instead.

For example, if processes in your flowchart are performed by resources named Clerk and Supervisor, you would place two Resource shapes and specify their characteristics (cost, capacity) by opening the property dialog boxes.

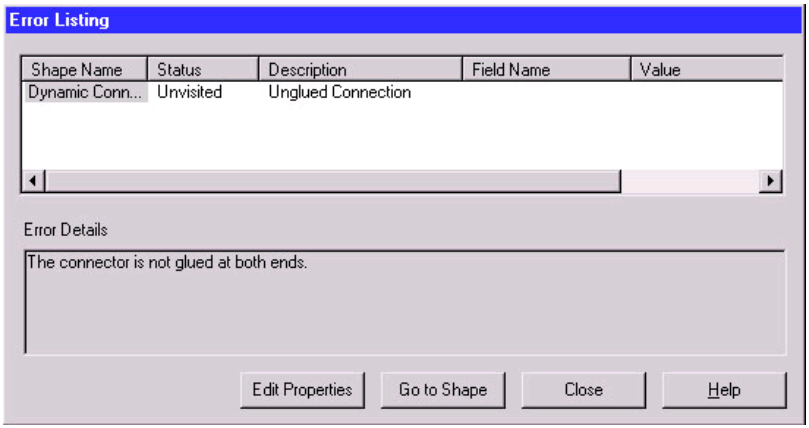
You only need to add the data shapes if you want to change any of the default values. Otherwise, they are not needed in the Visio drawing; they'll automatically be added to the spreadsheets when you transfer the drawing to Arena.

Simulate the process map in Arena

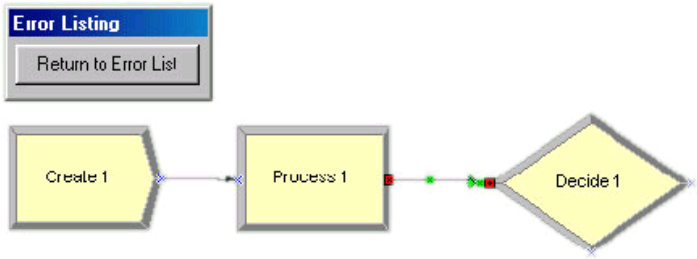
By using the Process Simulation stencil in Visio, you can define, document, and communicate your business processes. When you want to predict how your process will operate, examine it for bottlenecks, and compare possible changes to make informed decisions, you can transfer your process map from Visio to Arena, where the process will come to life with simulation and animation.

Check the drawing

While you're working with your process map, you can check it for any missing or incorrect data or connections. To check your drawing, use the **Check Drawing** item from the **Simulate** menu. After the drawing has been examined, a list of any warnings/errors is displayed, as shown below for a drawing that has a connector that is not glued at both ends.



If multiple problems are detected, each will be listed with a description and, in the case of invalid or missing properties, the field and value that are improper are listed. To correct the errors in your drawing, you can use the buttons at the bottom of the dialog box. For errors in shape properties, click the **Edit Properties** button, which displays the shape's dialog box, to correct the data. In the case of connector problems, click **Go to Shape**, which displays and selects the shape that caused the problem, as shown below. (Note that the Error Listing dialog box shrinks to allow more room for viewing the drawing.)



After you've corrected the problem, you can return to the list of errors/warnings by clicking the **Return to Error List** button. Once you finish correcting any additional problems, click the **Close** button. You can check your drawing at any time, correcting any or all errors that are detected.

Transfer the drawing to Arena

When you're ready to simulate your process, it's a simple matter of selecting the **Simulate Drawing** item from the **Simulate** menu. This tool opens Arena and translates your Visio

process map into an Arena simulation model. When the process is complete, you have an Arena model containing modules from the Basic Process panel that mirror the shapes you used from the Visio Process Simulation stencil, including the properties and connections you specified in Visio.

In Arena, you can set the simulation run length on the **Run > Setup** page. If you do not define the run length, you can stop the simulation at any time by clicking the **Pause** button, then the **End** button.

To run the simulation, click Arena's **Go** button. Your entities will animate the flowchart as the simulation proceeds, just as if you had created the model directly in Arena. Because the process is being simulated in Arena with the standard Basic Process modules, you have all of Arena's capabilities at your disposal, including reports on all aspects of your process via the Reports panel selections.

After you have simulated your process, you can save the Arena model (*.doe*) file to continue work in Arena, or you can close it to return to Visio for continued modifications. Because the transfer of the Visio drawing is from Visio to Arena (but not vice versa from Arena to Visio), it's usually best to return to Visio. However, you might want to save the Arena model in case you want to review the simulation or results as you're refining the Visio process map.

Simulate other Visio drawings

In addition to simulating process maps using the Process Simulation add-in, you can also simulate process maps created using other Visio stencils. Arena provides a shape map editor to define the relationship between the shapes in the Visio stencils and the modules in Arena's Basic Process panel. Arena's online help describes how to use this feature, which is accessed by opening the **Tools > Options** menu and clicking on the **Visio Options** tab.

Use Visio and Arena in your organization

The combination of Visio and Arena can widen and deepen the reach of process improvement in your organization. You can exploit the accessibility of Visio to ease the transition from drawing pictures to mapping processes. All types of process constituents—from those who govern the business to the staff who execute the process—can directly influence the definition of how the process works now … and what future "to-be" operations will provide the most dramatic improvements.

Distribute the Process Simulation add-in

The Process Simulation add-in is automatically installed with Arena and is available to all Visio users, with no additional fees or licenses required. You can distribute the add-in

throughout your organization so that everyone who has an interest in the process definition can access it directly through Visio.

With this approach, process maps can be created and shared throughout the organization on any desktop with a Visio license. When simulation and animation are needed to visualize and analyze the behavior of the process, the drawings can be transferred to systems with the Arena software. By using the Process Simulation template and checking the process maps as they're created, you can be confident that taking the step to move from static drawing to dynamic simulation will be seamless.

Publish process maps on a Web page

You can share your process maps—the drawing of your process flowchart—with others by using Visio's HTML export feature to publish on an intranet. To save the drawing in HTML format, select the **File > Save As** menu option in Visio and change the type to **`HTML Files`**. (Refer to Visio's online help for more information about the HTML export feature.)

The value of modeling and analyzing your processes can be further exploited by saving process performance information with the process maps. The results of simulation runs can be saved in HTML format, providing a benchmark against which actual operations or other improvement concepts can be compared. To save the reports, click the **Export** button when you have an Arena report open and select the **HTML** option as the file type.

Index

Numerics

2-way chance ▪ 32
2-way condition ▪ 32

A

Access module ▪ 79
Activate module ▪ 83
ActiveX support ▪ 131
Activity Area module ▪ 97
adding plots to the model ▪ 23
Advanced Process panel ▪ 49
  Advanced Set module ▪ 63
  Delay module ▪ 49
  Dropoff module ▪ 50
  Expression module ▪ 64
  Failure module ▪ 65
  File module ▪ 66
  Hold module ▪ 50
  Match module ▪ 52
  Pickup module ▪ 53
  ReadWrite module ▪ 53
  Release module ▪ 55
  Remove module ▪ 56
  Search module ▪ 59
  Seize module ▪ 57
  Signal module ▪ 60
  StateSet module ▪ 67
  Statistic module ▪ 68
  Storage module ▪ 70
  Store module ▪ 61
  Unstore module ▪ 61, 62
Advanced Set module ▪ 63
Advanced Transfer panel ▪ 71
  Access module ▪ 79
  Activate module ▪ 83
  Activity Area module ▪ 97
  Allocate module ▪ 83
  Convey module ▪ 80
  Conveyor module ▪ 92
  Distance module ▪ 95
  Enter module ▪ 71
  Exit module ▪ 81
  Free module ▪ 85
  Halt module ▪ 85
  Leave module ▪ 73
  Move module ▪ 86
  Network Link module ▪ 96
  Network module ▪ 96
  PickStation module ▪ 75
  Request module ▪ 87
  Route module ▪ 77
  Segment module ▪ 93
  Sequence module ▪ 90
  Start module ▪ 82
  Station module ▪ 78
  Stop module ▪ 82
  Transport module ▪ 88
  Transporter module ▪ 93
Allocate module ▪ 83
animating resources ▪ 22
animation ▪ 7, 19
  enhancing the graphics ▪ 21
animation scale factor ▪ 19
Arena Symbol Factory ▪ 128
  adding shapes to Arena ▪ 128
Assign module ▪ 35
Auto-Connect menu ▪ 8

B

Basic Process panel ▪ 7
  Assign module ▪ 11, 35
  Batch module ▪ 11, 33
  Create module ▪ 11, 27
  Decide module ▪ 11, 32
  Dispose module ▪ 11, 28
  Entity module ▪ 38
  Process module ▪ 11, 29
  Queue module ▪ 39
  Record module ▪ 11, 36
  Resource module ▪ 40
  Schedule module ▪ 42
  Separate module ▪ 11, 34

Set module ■ 44
Variable module ■ 41
Batch module ■ 33
Beta distribution ■ 109, 111
business process modeling ■ 131

C

Calendar Exceptions ■ 47
Calendar States ■ 48
chart ■ 7
collecting statistics ■ 36
Continuous ■ 112
Continuous empirical distribution ■ 109, 112
Convey module ■ 80
conveyor
definition for entity movement ■ 92
distance between stations ■ 93
flowchart modules ■ 79
movement ■ 79
status ■ 82
Conveyor module ■ 92
counter sets ■ 44
Create module ■ 7, 27
creation of entities ■ 27
Customer Support Center ■ 3

D

data modules ■ 11, 27, 38, 107
data objects ■ 11
Decide module ■ 10, 15, 32
decision-making processes ■ 32
defining model data ■ 12
Delay module ■ 49
Discrete empirical distribution ■ 109, 114
Dispose module ■ 10, 16, 28
disposing of entities ■ 28
Distance module ■ 95
Distributions
beta ■ 109, 111
continuous empirical ■ 109, 112
discrete empirical ■ 109, 114
Erlang ■ 109, 115
exponential ■ 109, 116
gamma ■ 109, 117
Johnson ■ 109, 118
lognormal ■ 109, 119
normal ■ 109, 120
Poisson ■ 109, 121
probability ■ 109
triangular ■ 109, 122
uniform ■ 109, 123
Weibull ■ 109, 124
document conventions ■ 2
documentation set ■ 2
drawing elements ■ 7
Dropoff module ■ 50
duplicating entities ■ 34

E

Enter module ■ 71
entities ■ 7
added to storage ■ 61
allocation of resources ■ 57
combining from different queues ■ 52
creating ■ 27
defining types ■ 38
disposing ■ 28
duplicating ■ 34
holding in queue ■ 50
moving from one group to another ■ 50
on conveyors ■ 79
releasing transporters ■ 85
removing from a given queue ■ 53
removing from queue and sending to a module ■ 56
removing from storage ■ 61
sequence flow through model ■ 90
time delays ■ 49
transfer to specified station ■ 77
transferring to station or module ■ 73
what are they? ■ 13
Entity module ■ 38
entity picture sets ■ 44
entity type sets ■ 44
Erlang distribution ■ 109, 115, 117
Exit module ■ 81
Exponential distribution ■ 109, 116
Expression module ■ 64

F

Failure module ▪ 65
File module ▪ 66
Flow module ▪ 102
Flow Process panel
  Flow module ▪ 102
  Regulate module ▪ 104
  Regulator Set module ▪ 107
  Release Regulator module ▪ 106
  Seize Regulator module ▪ 105
  Sensor module ▪ 101
  Tank module ▪ 99
flowchart modules ▪ 11, 27
flowchart objects ▪ 11
flowchart view ▪ 7
Free module ▪ 85

G

Gamma distribution ▪ 109, 117
grid ▪ 9
grouping within the model ▪ 33

H

Halt module ▪ 85
Hold module ▪ 50

J

Johnson distribution ▪ 109, 118

L

Leave module ▪ 73
Lognormal distribution ▪ 109, 119

M

Match module ▪ 52
model window ▪ 6
  flowchart view ▪ 7
  spreadsheet view ▪ 7
modules ▪ 6, 11
mortgage application example ▪ 5
Move module ▪ 86

N

Normal distribution ▪ 109, 120

O

online help ▪ 2
operating schedule ▪ 42

P

pause the simulation ▪ 19
PickStation module ▪ 75
Pickup module ▪ 53
picture libraries ▪ 127
  BasicProcess.plb ▪ 127
  listing of standard files ▪ 128
pictures
  changing from idle to busy ▪ 22
plots ▪ 23
Poisson distribution ▪ 109, 121
Probability distribution ▪ 109
process flowchart ▪ 7
process mapping ▪ 131
process maps ▪ 7
  checking the drawing ▪ 135
  publishing on a Web ▪ 138
  simulating other Visio drawings ▪ 137
  transferring the drawing to Arena ▪ 136
Process module ▪ 8, 13, 29
Process Simulation
  distributing the add-in ▪ 137
  stencil ▪ 132
Project Bar ▪ 6
Project Parameters dialog ▪ 18

Q

Queue module ▪ 39
queueing ▪ 33

R

ranking rules ▪ 39
reading data from input file ▪ 53
ReadWrite module ▪ 53
Record module ▪ 36
Regulate module ▪ 104
Regulator Set module ▪ 107

Release module ■ 55
Release Regulator module ■ 106
Remove module ■ 56
reports ■ 20
Request module ■ 87
resource constraints
  seizing and releasing ■ 29
Resource module ■ 17, 40
resource sets ■ 44
resource states ■ 40
resources
  defining states ■ 67
  definition ■ 40
  failures ■ 40
  releasing units previously seized ■ 55
  schedule definition ■ 42
Route module ■ 77
running the simulation ■ 24

S

sample models ■ 2
saving your model ■ 18
Schedule module ■ 42
Search module ■ 59
Segment module ■ 93
Seize module ■ 57
Seize Regulator module ■ 105
Sensor module ■ 101
Separate module ■ 34
Sequence module ■ 90
Set module ■ 44
sets definition ■ 44
Signal module ■ 60
simulating your process ■ 18
simulation run length ■ 137
SMARTs library ■ 2, 12, 15
snap ■ 9
split view ■ 21
splitting batches ■ 34
spreadsheet view ■ 7
Start module ■ 82
StateSet module ■ 67
Station module ■ 78
Statistic module ■ 68
statistics, defining collection ■ 68
stepping through the simulation ■ 19
Stop module ■ 82
Storage module ■ 70
Store module ■ 61

T

tally sets ■ 44
Tank module ■ 99
time delay ■ 13
Time Pattern ■ 45
time patterns ■ 45
transfer an entity ■ 77
Transport module ■ 88
transporter
  assigning to an entity ■ 83
  capacity activation ■ 83
  changing to inactive status ■ 85
  moving from station to station ■ 86
Transporter module ■ 93
transporter modules ■ 83
Triangular distribution ■ 109, 122

U

Uniform distribution ■ 109, 123
Unstore module ■ 61

V

Variable module ■ 41
variables
  assigning new values ■ 35
  definition ■ 41
viewing reports ■ 20
Visio ■ 131
Visio Process Simulator ■ 131
  adding data shapes ■ 135
  entering custom properties ■ 134
  placing and connecting shapes ■ 133
  simulating the process map ■ 135
  starting a drawing ■ 131

W

Web support ■ 3
Weibull distribution ■ 109, 124
writing data to output device ■ 53